3-5-2024 2:00 PM

Investigating Tree- and Graph-based Neural Networks for Natural Language Processing Applications

Sudipta Singha Roy, *Western University*

Supervisor: Mercer, Robert E., *The University of Western Ontario*

A thesis submitted in partial fulfillment of the requirements for the Doctor of Philosophy degree in Computer Science

Abstract

In the information explosion era, the ability to automatically extract knowledge and gain insights from diverse linguistic genres has become imperative. Comprehending intricate linguistic expressions constitutes an indispensable facet of artificial intelligence. Deep learning techniques have emerged as powerful tools for classification, relation extraction, semantic similarity measurement, and document summarization, offering the promise of revolutionizing our understanding of these crucial domains. In the dynamic landscape of Natural Language Processing (NLP), the integration of syntactic and semantic elements stands as a pivotal frontier. This investigation delves into incorporating both syntactic and semantic dimensions within NLP applications. By leveraging tree- and graph-based neural networks, this study pioneers a holistic approach that augments language understanding and processing capabilities. Through the fusion of structural and semantic-driven insights, this work tries to explore various NLP applications for two linguistic genres: scientific text, and psycho-linguistic texts. Scientific articles inherently embody a sophisticated framework of information representation, necessitating a depth of background knowledge for comprehension. This requisite background knowledge is gleaned through a meticulous examination of the citations interwoven within the ongoing paper. The objective of this endeavor is to scrutinize the citation linkage task, serving as an avenue for extracting the essential background information imperative for the meticulous analysis of scientific documents. Furthermore, for summarization, the citation network is leveraged to augment the performance of summarization models by furnishing additional contextual underpinnings. Different tree-structured neural networks are systematically explored to discern relations between various biomedical entities within scientific articles, thus contributing to the efficacy of relation extraction tasks. In the contemporary landscape dominated by the proliferation of social media, natural language processing emerges as a potent instrument for psychologists to delve into the analysis of individuals' personality traits. Conventional models, hampered by their incapacity to grapple with extended textual sequences exceeding their token intake limit, encounter limitations. This work propounds innovative solutions through the utilization of tree-structured neural networks and graph attention networks, facilitating the identification of personality traits from protracted written compositions.

Keywords: Semantic Similarity, Syntactic Representation, Tree-structured Transformers, Graph Attention Network, Heterogeneous Graph Network, Citation Linkage, Protein-Protein Interaction, Drug-Drug-Interaction, Multi-head attention, Multi-branch attention, Scientific Article Summarization, Personality Trait Identification, Text Classification.

Summary for Lay Audience

In today's age of information overload, it's crucial to automatically extract insights from diverse types of textual representaitons. This is especially important for artificial intelligence to comprehend complex linguistic expressions. Deep learning techniques, powerful tools for tasks like classification, relation extraction, and document summarization, hold the potential to revolutionize our understanding of these crucial domains.

Within the dynamic field of Natural Language Processing (NLP), integrating both syntactic and semantic elements is a key frontier. This research explores the combination of these dimensions using tree- and graph-based neural networks, offering a holistic approach to enhance language understanding and processing capabilities. The study focuses on two linguistic genres: scientific text and psycho-linguistic texts.

Scientific articles are intricate in their representation of information, requiring a depth of background knowledge for comprehension. This research meticulously examines citations within scientific papers to extract essential background information. The primary goal is to scrutinize citation linkages, providing necessary context for the detailed analysis of scientific documents. Additionally, the citation network is utilized to improve summarization models by adding contextual underpinnings along with a reflection of the research community's view. Various tree-structured neural networks are systematically explored to discern relations between biomedical entities within scientific articles, enhancing the efficacy of relation extraction tasks.

In today's world dominated by social media, natural language processing becomes a powerful tool for psychologists studying individuals' personality traits. Conventional models face limitations in handling lengthy textual sequences. This work introduces innovative solutions using tree-structured neural networks and graph attention networks to identify personality traits from extended written compositions. These approaches aim to overcome the challenges posed by the token intake limit of traditional models, providing new avenues for understanding and analyzing complex human expressions.

Co-Authorship Statement

This thesis is presented in an integrated article format, comprising a total of 11 papers. All the papers related to this thesis are either published, in press, or planned for submission. The primary authorship of all but one paper is attributed to the author of this dissertation. Sections 4.1, 4.2, 5.1, 5.2, 5.3, 6.3, 7.1, 7.2, and 7.3 represent collaborative endeavors involving Sudipta Singha Roy and Dr. Robert E. Mercer. Sudipta Singha Roy, the author of this dissertation, undertook responsibilities ranging from the literature survey, approach development, study design formulation, and research implementation to manuscript drafting for publication. Serving as the research supervisor, Dr. Robert E. Mercer played a crucial role in the study's design, result interpretation, and manuscript preparation for publication.

In Section 6.2, Souvik Kundu, the third co-author and former MSc graduate student in Dr. Mercer's lab, assisted in processing the manuscript.

In Section 6.1, Amirmohammad Kazemini, the primary author and former MSc graduate student in Dr. Mercer's lab, originated the concept of interpretable representation for personality trait identification and conducted an extensive literature review. Sudipta Singha Roy, the secondary author of this manuscript and the author of this dissertation, contributed to the design of the models and implementation. Dr. Robert E. Mercer and Dr. Erik Cambria, Professor of Computer Science and Engineering at Nanyang Technological University, Singapore, guided the research work with their knowledgeable supervision and assisted in manuscript preparation for publication.

Acknowledgements

I extend my heartfelt gratitude and appreciation to my supervisor, Dr. Robert E. Mercer. He stands as one of the most exceptional mentors I have encountered. His unwavering enthusiasm for my ideas, acknowledgment of our accomplishments, and continuous encouragement throughout the research journey have been invaluable. Notably, his constant positivity, always accompanied by a smile, has made the research experience not only fruitful but also enjoyable. Under his guidance, I have grown as an independent researcher, and his insightful methods have shaped the completeness of my scholarly pursuits. In challenging moments, his steadfast support has been my anchor. His optimism played a pivotal role in lifting me up during times of adversity. He has consistently served as my mentor and guardian, offering a supportive presence with whom I could openly share all my challenges and concerns. I am profoundly grateful for his guidance, and I acknowledge that I can never fully repay the debt of gratitude I owe him.

I convey my deep appreciation and thanks to my parents for their unwavering support and enduring faith in me. My mother, in particular, has been a perennial source of inspiration. It is her unyielding encouragement that propelled me to embark on the journey of pursuing a PhD. In 1976, owing to familial responsibilities, she had to defer her own pursuit of a PhD. Witnessing her unfulfilled dream, it became my aspiration to carry the torch forward and achieve this milestone for both of us.

I would also like to express my deep appreciation for my nephew, Souvik. Since his arrival in Canada, my life has taken on a new vitality, reminiscent of the joyous moments of the past. Souvik has been a constant pillar of support, standing by me through thick and thin, making every situation more manageable.

I extend my heartfelt gratitude to my dear friends, Navid and Anurag, whose unwavering support and companionship have been invaluable to me. Their presence has added warmth and camaraderie to my journey. I would also express my gratitude to Xindi and Anemily for their invaluable insights and suggestions shared during our lab meetings and discussions, which greatly enriched my work.

In conclusion, I dedicate this work wholeheartedly to my beloved daughter, Sam. Despite the physical distance that separates us, she remains the light of my life. Her presence, even from afar in Bangladesh, is my source of inspiration. The radiance of her smile has the power to dispel all hardships and stress, serving as a constant reminder of the profound joy she brings to my world.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Thesis Statement

Natural Language Processing (NLP) and deep learning have become indispensable in today's technological landscape, revolutionizing how we interact with machines and process vast amounts of textual data. NLP enables computers to understand, interpret, and generate human language, powering applications such as chatbots, language translation, and sentiment analysis. Deep learning, particularly through neural networks, has proven to be a game-changer in enhancing the performance of NLP tasks. The ability of deep learning models to automatically learn hierarchical representations of data has significantly improved the accuracy and efficiency of language-related applications.

In contemporary research landscapes, the amalgamation of Natural Language Processing (NLP) and deep learning has emerged as a transformative force, particularly in the realms of scientific literature analysis and psychological trait identification. NLP facilitates the rapid and comprehensive analysis of vast scientific texts, accelerating the pace of literature review and knowledge synthesis. Through deep learning techniques, such as recurrent neural networks and attention mechanisms, these systems can discern intricate patterns, identify key concepts, and even predict emerging trends in scientific research.

In the field of psychological trait identification, NLP-powered tools contribute significantly to the analysis of textual data related to individual behavior and personality traits. Deep learning models excel at discerning subtle nuances in language, allowing for the extraction of valuable insights from written expressions. This capability proves invaluable in psychological research, aiding professionals in identifying patterns associated with specific traits, emotions, or mental health indicators. The integration of these technologies not only expedites the analysis process but also enhances the depth and accuracy of understanding human behavior, paving the way for more nuanced and personalized interventions in mental health and well-being. As the

volume of scientific literature and the complexity of psychological data continue to grow, the synergy between NLP and deep learning stands as a cornerstone in advancing our comprehension of both scientific knowledge and the intricacies of the human mind.

For the NLP tasks, the incorporation of both semantic and syntactic information is paramount for deep learning models, as it addresses the inherent complexities of language comprehension. Syntactic structures, representing the grammatical relationships within sentences, provide a fundamental scaffold for understanding how words interconnect. By integrating syntactic information, deep learning models can grasp the hierarchical and sequential nature of language, enhancing their ability to discern context and meaning. On the other hand, incorporating semantic understanding enables deep learning models to capture the nuances, ambiguity, and context-dependent meanings that are inherent in human language. This is particularly significant in the tasks mentioned above, where the accurate interpretation of meaning is pivotal for successful outcomes. Moreover, the synergy between semantic and syntactic information offers an holistic approach to language representation, empowering deep learning models to move beyond surface-level analysis. In essence, the joint incorporation of semantic and syntactic information is a key enabler for advancing the capabilities of deep learning models in NLP tasks, ultimately bridging the gap between human-like language comprehension and machine-based language processing.

In this study, the focus has been investigating four NLP tasks, semantic similarity measurement, relation extraction, classification, and document summarization, for two linguistic genres: scientific literature and psychological statements. Most of the NLP tasks and the state-of-the-art models consider preserving either semantics of the text or syntactical information. In this study, we have tried to make a bridge between these two backbone parts of NLP by introducing different models for the above mentioned tasks.

1.2 Problem Statement

Natural Language Processing (NLP), a pivotal subfield of artificial intelligence, focuses on endowing machines with the capacity to understand, interpret, and produce human language. It encompasses a diverse range of applications that are reshaping our interactions with and extraction of insights from textual data [100].

Prior to the widespread integration of deep learning methodologies, NLP applications predominantly leaned on rule-based systems and conventional machine learning algorithms. In the pre-deep learning phase, tasks like machine translation, sentiment analysis, and named entity recognition primarily relied on manually crafted rules and intricate feature engineering. These methodologies encountered challenges in capturing the intricate subtleties of language

and faced limitations in effectively handling diverse and extensive datasets.

The landscape of NLP underwent a revolutionary transformation with the advent of deep learning techniques. Models like recurrent neural networks (RNNs) [194], long short-term memory networks (LSTMs) [84], and attention mechanisms [18, 127, 223] demonstrated exceptional efficacy in unraveling complex linguistic patterns and dependencies. The emergence of potent transformer [223] architectures, as evidenced by models such as BERT (Bidirectional Encoder Representations from Transformers) [56] and Longformer [22] have marked a paradigm shift in NLP. These models, trained on extensive datasets, have acquired the ability to grasp contextual language representations, surpassing traditional methods across various tasks. The application of deep learning in NLP has significantly bolstered the precision and scalability of systems. Tasks that were previously arduous, such as summarization, relation extraction, and sentiment analysis, now benefit from the context-aware capabilities inherent in deep learning models. Furthermore, the development of expansive pre-trained language models has streamlined transfer learning, allowing for fine-tuning on specific tasks with more limited datasets [102]. This trajectory of NLP evolution from rule-based and traditional machine learning approaches to the integration of deep learning methodologies has ushered in a new era characterized by more robust, context-aware, and scalable natural language processing applications.

Though the current research utilizing deep learning models has shown prominent improvement in NLP tasks, still, the majority of works ignore linguistic features that could be integrated. In contrast to other endeavours, NLP is a complex task to perform due to the phrasal representation of words and the dependencies between non-sequential words which are placed at a distance in the sentences [118]. The sequential deep learning models and the large language models ignore this structural information of the texts. In an effort to address this lacuna, our study endeavours to construct models for different NLP applications that incorporate linguistic features with an objective to enhance the performance of these models by preserving more robust semantic representations. In this study, we have investigated four NLP applications: classification, relation extraction, semantic similarity measurement, and document summarization utilizing tree-structured neural networks [6, 118] and graph attention networks [224, 227]. We have explored these tasks covering two different genres: scientific and psychological texts. For investigating the classification models we have experimented with psychological texts which identify personality traits of individuals from social media posts and question-answer sets. We have revisited the protein-protein and drug-drug interaction identifications from biomedical research articles to explore the essence of the relation extraction task. To ease the readers while going through any research article we have formulated the citation linkage problem as a semantic similarity task where the investigated models fetch the semantically similar state-

ments from the reference articles that correspond to the citing statements from the ongoing paper to provide additional background information necessary to grasp the concept presented in the ongoing paper. Finally, we have investigated two approaches of the scientific article summarization task: extractive and abstractive summarizations. The summarization models investigated in this work incorporate the citation network to provide the background information and the impact of the article in the research community. Furthermore, we have investigated a joint training method of the extractive and abstractive summarizers and a semantic-induced loss function so that each counterpart's performance is improved.

1.3 Motivation

Within the dynamic domain of Natural Language Processing (NLP), the profound import of syntactic and semantic linguistic features remains indubitable. These pivotal elements serve as the bedrock upon which NLP tasks can be built, unlocking the door to a deeper understanding of human language and enabling machines to navigate the intricacies of communication with increasing finesse.

Syntactic features, encapsulating the structural arrangement of words and the relationships between them, act as the grammar framework of language. They provide a roadmap for deciphering the intricate tapestry of sentences, facilitating the extraction of meaning from the syntactic structures that underpin human expression. As NLP methods grapple with the nuances of syntax, they gain the ability to comprehend not only the literal meaning of words but also the intricate dance of grammar that imparts layers of subtlety and context to language.

On the other hand, semantic attributes plunge into the realm of meaning, navigating the intricate nexus of associations and connotations that endow words with significance. Semantic understanding goes beyond the surface-level interpretation of individual words, encompassing the contextual nuances and relationships that define the true essence of communication. Harnessing semantic features empowers NLP systems to discern subtle nuances, grasp metaphorical expressions, and navigate the rich tapestry of language with a depth that mirrors human comprehension.

Together, the integration of syntactic and semantic features forms a formidable synergy that elevates NLP tasks to unprecedented levels of sophistication. Whether it be text summarization, personality trait identification, relation extraction, or semantic similarity measurement, a nuanced grasp of syntax and semantics equips NLP models with the cognitive tools to unravel the complexity of human language. In essence, the journey toward natural language understanding is paved by the profound interplay of syntactic and semantic features, propelling NLP into realms of comprehension that echo the intricacies of human communication [6, 118]. That

is why incorporating syntactic information to preserve better semantics for the NLP tasks has been the main motivation of this study. Driven by this imperative, our scholarly pursuit has been devoted to scrutinizing various deep learning models across two distinct genres, enriched by the integration of tree-structured information representations. These neural networks, configured in a tree structure, endow the scrutinized models with an augmented repository of syntactic details, a facet notably absent in extant works.

Another facet of our comprehensive inquiry is dedicated to the refinement of summarization techniques specifically tailored for scientific articles. The summarization of scientific content introduces distinct complexities when compared to articles of other genres. Notably, scientific documents, characterized by their substantial length, introduce unique challenges to the summarization process. Furthermore, a comprehensive grasp of these documents necessitates background information, adding another layer of intricacy to the summarization process. Moreover, the dynamic nature of research impact introduces a temporal dimension. Enabling summaries to encapsulate evolving perspectives can be a help for the reader.

Recognizing these intricacies as focal points, our research is motivated by an aspiration to augment neural network summarizers designed for scientific articles. A key innovation in this pursuit involves integrating citation networks to provide essential background information and encapsulate the evolving perspectives within the corresponding research communities. This augmentation not only addresses the contextual challenges inherent in scientific articles but also positions summarizers as invaluable tools capable of reflecting the evolving impact of research work within the pertinent scholarly societies. In response to the inherent lengthiness of scientific articles, we have delved into the exploration of segmentation mechanisms as a concurrent task. This multifaceted approach underscores our commitment to comprehensively address the challenges unique to the domain of scientific article summarization.

1.4 Objectives

For a resolution of any quandary within the NLP applications, it becomes imperative to cultivate a potent representation of the subject of scrutiny, whether it be a singular word, an entire sentence, or the entirety of a document. Contemporary paradigms in machine learning and deep learning bestow the capability to distill abstract features from data, presenting users with a methodological framework to craft robust architectures. These architectures harness well-defined differentiable objective functions, meticulously mapping inputs to outputs.

However, our purview extends beyond the conventional, as we aspire to transcend prevailing benchmarks in the intricate NLP tasks. Our approach is anchored in the amalgamation of machine learning models and profound linguistic acumen, a convergence designed to propel

the frontier of what is considered state-of-the-art in addressing the complexities inherent in NLP challenges. In a more granular context, our focus centers on enhancing the efficacy of deep learning models across diverse NLP applications involving the integration of syntactic information and the preservation of heightened semantic precision, achieved through the incorporation of contextual information. As the exposition advances, it will become apparent to the reader that we steadfastly adhere to the following objectives throughout the course of this thesis.

To incorporate the syntactic information in the downstream NLP applications. We have designed some deep learning models that incorporate the phrasal and inter-word dependency information by using constituency and dependency tree-structured neural networks. Combined with an attention mechanism, these models attain the capability to provide better semantics in the sentence- and document-level representations.

To enrich word representations with task-specific context information. We have introduced one word refinement module for the downstream tasks to generate enriched word representations having task specific context information. Our proposed deep learning models with these enriched word representations have shown prominent performance improvement for various NLP tasks.

To enhance the performance of the summarization models utilizing a citation network. We have designed summarization models that intake background information regarding the considered scientific articles using a citation network. These models also reflect the impact of the considered articles incorporating citing statements in the summaries as well.

To improve the quality of the summarizers by introducing a semantic-induced training mechanism. We have introduced a joint training of extractive and abstractive summarizers with a semantic-induced loss function. This training mechanism has improved the performance of the summarizer models in terms of the performance metrics.

1.5 Contributions

The thesis unfolds a tapestry of significant contributions, a mosaic of insights that may appear diverse given the expansive terrain within the NLP domain. However, within this diversity lies a coherence meticulously woven by overarching themes. In the ensuing discussion, we delineate the threads that converge these contributions into cohesive motifs, offering a succinct preview of their accomplishment and the valuable insights they furnish. A more detailed exposition of these details will be unveiled in subsequent chapters.

Structural Information Enhancement We embark on a comprehensive exploration of various NLP applications across two distinct genres. Our focus lies in enriching word, sentence,

and document representations by incorporating structural information, ultimately elevating the overall model performance.

Citation Linkage Framework A bespoke framework and silver standard corpus are introduced to facilitate the understanding of ongoing research articles. This framework addresses the citation linkage task, reformulated as a semantic similarity measurement challenge. The model retrieves semantically similar sentences from reference articles based on the citing statements in the ongoing paper.

Biomedical Relation Extraction Models Tree-structure-based relation extraction models are developed to discern interactions between diverse biomedical entities. These models, augmented with additional phrasal and inter-word dependency information, outperform their counterparts, establishing superior performance.

Personality Trait Identification as Multi-Label Classification The intricacies of personality trait identification are approached as a multi-label document classification problem. In contrast to token limitations in other BERT-based models, our proposed models overcome these constraints, accommodating texts of variable lengths.

Scientific Article Summarization Augmented by Citation Networks Performance in scientific article summarization is enhanced by integrating additional background information through citation networks. Our models not only produce improved summaries but also reflect the impact of articles on the corresponding research society. An in-house summarization corpus, comprising 10k research articles and citation information, is introduced for this purpose.

Semantic-Induced Joint Training for Extractive and Abstractive Summarizers We introduce a novel approach of semantic-induced joint training for extractive and abstractive summarizers, resulting in improved individual performance. Furthermore, the incorporation of segmentation and citation linkage into the summarization task contributes to a substantial performance boost.

Word Refinement Module for Downstream NLP Tasks A word refinement module is introduced to enhance downstream NLP tasks by allowing models to produce superior sentence and document representations. This module updates word embeddings based on context, aligning with the BERT fine-tuning principle while demanding less computational resources.

Last but not least, our models mark substantial quantitative advancements, a testament to their efficacy in the landscape of NLP. Rigorous experimentation underscores the competitive edge inherent in each model, showcasing their prowess when benchmarked against the existing state-of-the-art models spanning diverse tasks. Our empirical endeavours transcend convention as we elevate the performance bar to redefine the state-of-the-art across a spectrum of NLP tasks. These tasks encompass document classification, relation extraction, semantic similarity measurement, and the nuanced domain of scientific document summarization.

1.6 Thesis Organization

The thesis is structured to show the evolution of our ideas, in the thesis as a whole and in each of the chapter. The thesis is rooted in our desire to introduce syntactic and semantic information into our deep learning models, starting with the citation linkage task and continuing through the following applications, creatively building on the techniques used in the prior applications.

Chapter 1 commences with an introductory exploration of the thesis. This initial chapter delineates the core problem under investigation, articulates the motivations propelling this research endeavour, elucidates the overarching objectives, and underscores the contributions poised to enrich the broader research community.

Chapter 2 encapsulates the benchmarks associated with our carefully examined experiments for four different NLP applications. This chapter briefly describes the prominent works that have been explored before addressing the tasks we have investigated here.

Chapter 3 delves into the performance metrics utilized across various NLP applications, providing equations and explanations for these metrics.

Chapter 4 explains the citation linkage problem, how it has been formulated as a semantic similarity measurement problem and how it has been tackled. Section 4.1 introduces the proposed silver corpus for the citation linkage task, detailing corpus creation methods and validation by human annotators. Section 4.2 explains the construction of the citation linkage framework to establish a connection between the citing sentence and the referenced text span within the cited biomedical research article. It involves the integration of syntactic information by means of utilizing constituency and dependency tree-structured neural networks for the semantic similarity measurement task.

Chapter 5 shifts our focus from semantic similarity to relation extraction. Section 5.1 investigates the performance of different tree-structured neural networks and ensembles of them for the relation extraction task, focussing on identifying the interactions between different proteins in biomedical text. Section 5.2 introduces a word refinement module using an heterogeneous graph attention network. The word refinement module provides a separate sentence representation which is concatenated with the sentence representations from the tree transformers to provide a better sentence representation for the following classifier layer. Section 5.3 modifies this idea by having the word refinement module update the word embeddings for the tree-transformers rather than generating a separate sentence embedding. The model has achieved state-of-the-art performance for the protein-protein and drug-drug interaction identification tasks across all the benchmark corpora. These investigations laid the groundwork for the inception of a task-specific word refinement module, initially applied at the sentence level and subsequently extended to paragraph-level applications in Chapter 6 Section 6.3.

Chapter 6 delves into our endeavours related to personality trait identification. Initially, Section 6.1 deals with the personality trait analysis as a semantic similarity measurement task and measures the similarity of individuals' statements against the baseline statements in the semantic space. Section 6.2 continues with personality trait analysis, formulating it as a multi-label document classification problem. The two layered hierarchical approach used here overcomes the 512-token limitation of the previous BERT-based state-of-the-art models. The model encodes sentences using tree-transformers and then a graph attention network accumulates the sentence embeddings. Section 6.3 enhances this idea by incorporating a statement-to-sentence and a sentence-to-word refinement module. These refinement modules generate context-enriched word embeddings so that the sentence and statement encoders can generate representations that better preserve semantics. This final model achieves state-of-the-art performance.

Chapter 7 discusses our exploration of text summarization of scientific articles, encompassing both extractive and abstractive techniques. A significant stride was made by leveraging citation networks to augment summarizer units. Section 7.1 introduces the corpus we have created for the scientific document summarization by integrating the citing statements. A large language model-based abstractive summarizer and a graph attention network-based extractive summarizer are trained in parallel to improve each summarizer's performance. Section 7.2 extends the previous summarizer by enabling the model to utilize both sides of the citation network: the reference side and the citing side. Finally, Section 7.3 introduces a novel training mechanism for summarization models, jointly training extractive and abstractive summarizers with a semantic-induced loss function, demonstrating improved performance for individual summarizer units.

Finally, **Chapter 8** summarizes the thesis, delving into its inherent limitations and suggesting potential avenues of extension and further exploration.

Chapter 2

Related Work

This chapter provides a succinct overview of the related work in the corresponding application domain. Chapters 4 to 7 comprehensively delve into the related work pertinent to its respective domain, offering readers a more nuanced understanding of the existing literature and the contextual landscape in which the research is situated.

Our investigations encompass scientific document analysis, wherein we have delved into the citation linkage task, scientific article summarization (embracing both extractive and abstractive methods), and relation extraction from biomedical research articles, whereas for psychological text analysis we have investigated the personality trait identification task as a semantic similarity measurement problem at first, and then as a multi-label classification problem.

Regarding the citation linkage task, we have adopted a methodology grounded in semantic similarity measurement. In the subsequent exposition, various models for semantic similarity measurement are expounded upon.

Following this, our endeavours extended to discerning relationships within biomedical entities from scientific articles, specifically between proteins (PPI) and drugs (DDI). Despite the potential for protein or drug relationships spanning multiple sentences, the experiments showcased herein are confined to identifying these associations within a singular sentence. This limitation stems primarily from the annotation structure of the benchmark corpora at our disposal. For the PPI and DDI tasks, we have formulated solutions as relation extraction tasks. The ensuing section provides a succinct overview of the methodologies employed in tackling the intricacies of PPI and DDI problems.

A pivotal facet of the scientific literature analysis conducted in this study pertains to distilling the essence of the literature. Scientific document summarization presents two distinct approaches: extractive and abstractive summarization. Noteworthy challenges in summarizing scientific literature include the extended length of documents, the intricacies of information structure, and the prerequisite for background knowledge. In our research, we endeavoured to

surmount these challenges through diverse methodologies. Our efforts extended to conducting experiments involving the parallel training of extractive and abstractive summarizers, fostering mutual guidance to enhance the efficacy of individual summarization units. Additionally, we sought to capture the enduring impact of scientific articles on their respective research communities over time, utilizing citation networks to afford readers insights into the application of the proposed methods within the considered paper's domain. This section also delves into noteworthy research contributions within this domain.

For psychological text analysis, our efforts have been directed toward discerning distinct personality traits from textual content and our work has investigated two personality trait models. These endeavours were conducted using three widely utilized benchmark corpora. The task of identifying personality traits was approached through two distinct methods: gauging the semantic similarity of provided statements in relation to benchmark statements, and employing multi-label classification techniques leveraging tree and graph-based neural networks. Notably, in contrast to the sentence-level focus inherent in the classification tasks associated with scientific literature analysis, the tasks involving the identification of personality traits have been executed at the paragraph level. The concluding segment herein provides an overview of prior contributions within this field, encapsulating the antecedent endeavours and achievements in the domain under consideration.

2.1 Related Work: Scientific Article Analysis

2.1.1 Citation Linkage

The examination of citations in the realm of scientific research has given rise to extensive scholarly endeavours. Citation analysis seeks to discern the specific section within a referenced article to which a given sentence pertains, encompassing elements such as the abstract, introduction, methodology description, and results analysis [70, 71]. Nevertheless, this form of inquiry encounters limitations in pinpointing the expanse of citation spans.

An alternative avenue of research focuses on delineating the citation span. PolyU, for instance, employed RankSVM over sentence chunks to predict the span of cited text [39]. Baruah et al. [19] computed cosine similarity of word embeddings for the citation linkage task, while Yeh et al. [253] employed majority voting across various machine learning classifiers, considering lexical, knowledge-based, corpus-based, syntactic, and surface features.

The CL-SciSumm Shared Task endeavours to address three facets: locating the span of cited text in the citation sentence ("citance"), identifying the discourse facet of the cited sentence, and summarizing the referenced article using only frequently quoted text spans from the

document [134]. However, the latter two sub-tasks lie beyond the purview of the present work. Ma et al. [134] employed diverse classifiers and a voting mechanism over similarity, rule, and position-based features for determining the similarity between citing and cited statements in CL-SciSumm-17. The linkage between citing and cited sentence pairs was established by Li et al. [121] using inverse document frequency and Jaccard similarity. Subsequent works by the same authors involved the computation of sentence vectors through the concatenation of 200-dimensional word vectors [119], followed by the application of a convolutional neural network (CNN) over the concatenated vector representation [122]. In both instances, the cited text span was ascertained by measuring cosine similarities between citing and candidate cited statements. Other researchers, such as AbuRa'ed [2], have also engaged with the CL-SciSumm corpus.

More recently, BERT-based models have become prevalent in the citation linkage task, featuring prominently in numerous experiments. Gidiotis et al. [72] fine-tuned BERT to identify the referenced cited sentences from the cited document. Zerva et al. [259] applied a CNN over SciBERT-based features [21] to determine the specific text span in the cited article being referenced. They augmented features from the BERT-based model for comprehensive feature generation. Umapathy et al. [220] leveraged key-phrase similarity using the Rapid Automated Keyword Extraction Algorithm [187] and a BERT-based architecture for cited text span identification.

Nevertheless, a paucity of citation linkage studies exists for biomedical research papers, a domain characterized by diverse representations of identical components. A noteworthy contribution in this domain dates back to 2017, where Houngbo and Mercer [85] applied a traditional machine learning approach over their proprietary, albeit modestly sized, expert-annotated corpus, representing the singular human-annotated corpus for the citation linkage task in the biomedical domain thus far.

2.1.2 Relation Extraction between Biomedical Entities

Numerous Natural Language Processing (NLP) methodologies have emerged to discern connections among proteins. Initially, pattern-based techniques prevailed, relying on syntactic and lexical features to establish rules for relationship identification [27, 114]. However, these models faced challenges in accurately handling intricate relationships expressed in relational and coordinating clauses. In contrast, dependency-based methods, with a more concentrated emphasis on syntax, offer versatility across a broader spectrum of scenarios [60, 150].

Kernel-based techniques constitute another prevalent approach for identifying correlations between proteins, leveraging rich structural information acquired through dependency struc-

tures and syntactic parse trees [204]. Airola et al. [7] proposed a method examining information from linear and dependency subgraphs to identify interactions between target proteins. Miwa et al. [149] introduced a system incorporating a Support Vector Machine with weighted feature vectors derived from multiple corpora. Kim et al. [109] utilized a walk-weighted sub-sequence kernel by matching e-walks and v-walks on the shortest dependency path. Zhang et al. [260] developed a neighborhood hash graph kernel-based model for extracting Protein-Protein Interactions (PPIs), while Chang et al. [42] used a convolution tree kernel and PPI patterns to extract interlinkages between proteins. Murugesan et al. [152] proposed the distributed smoothed tree kernel, showcasing substantial advancements in comparison to other kernel methods for this task.

The advent of deep learning models has ushered in a plethora of experiments to unravel PPI relationships from biomedical literature [88, 175, 267]. Zhao et al. [270] pioneered the application of deep learning in PPI relation extraction, employing an autoencoder on unclassified training data to prepare parameters for a multi-layer perceptron (MLP) model. Peng et al. [164] utilized a double-channel CNN, with one channel incorporating syntax-based features and the second channel applying convolution based on parent word information. For PPI extraction, Zhang et al. [267] implemented a three-channel CNN, incorporating convolution operations on original words, positional encoding, the shortest dependency path, and encoding features for dependency relations in each channel.

Subsequent studies explored the efficacy of Recurrent Neural Networks (RNNs) in processing sequential data for the PPI task [4, 64, 88, 244, 245]. Hsieh et al. [88] concatenated output vectors from a Bi-LSTM, fed with sentence input, to generate a sentence vector representation. Yadav et al. [244] utilized the shortest dependency information between unit pairs as input to a Bi-LSTM with structured attention. Yadav et al. [245] introduced a self-attentive approach for simultaneous tasks: extraction of protein-protein interactions and extraction of drug-drug interactions. Ahmed et al. [4] applied structured attention over dependency tree-LSTMs, demonstrating the superiority of tree-structured neural networks over sequential models.

Apart from PPI, another task we have investigated for relation extraction is identifying relation between drugs (DDI). Fei et al. [64] investigated a graph-based approach operating on a fully connected graph composed of either word or phrase nodes. This approach is designed to leverage the structural information present in the data for improved performance in Protein-Protein Interaction (PPI) and Drug-Drug Interaction (DDI) extraction. Asada et al. [14] utilized molecular structure and drug descriptions for retrieving DDIs, while Gu et al. [80] fine-tuned PubMedBERT to extract relations between drugs. Following this, Asada et al. [15] employed a knowledge graph with PubMedBERT for the DDI extraction task, demonstrating an integrated approach for enhanced knowledge representation and extraction performance.

2.1.3 Scientific Document Summarization

In light of notable progress in short document summarization, there has been a burgeoning interest in long document summarization, particularly for substantial content such as scientific articles. Researchers are exploring both extractive [189, 227, 242] and abstractive [8, 174, 228] approaches.

Extractive Text Summarization (ETS) aims to classify sentences in a document, determining whether a specific sentence should be included in the summary. Recent ETS models for lengthy documents predominantly rely on transformer-based architectures [22, 258], chosen for their ability to handle longer sequences compared to RNN-based models. BERTSUMEXT [126] fine-tunes BERT with stacked Transformer layers and a sigmoid classifier. HIBERT [264] introduces a hierarchical Transformer encoder with pre-training and fine-tuning specifically for ETS. The siamese-BERT architecture is utilized by MatchSum [271] to select candidate extractive summaries based on semantic similarity. State-of-the-art extractive summarizers for scientific documents include HiStruct+ [189], GRETEL [242], HEGEL [261], and Lodoss [47]. HiStruct+ [189] innovatively incorporates hierarchical structure information into an extractive summarization model, leveraging both local and global contextual details. It is based on a pre-trained Transformer language model, aligning with the principles of BERTSUMEXT [126]. A significant contribution is the introduction of hierarchical positional encoding for sentences, facilitating the integration of hierarchical information within Pre-trained Language Models (PLMs) for the summarization task. GRETEL is a significant advancement in Extractive Text Summarization (ETS) for long documents, combining graph contrastive topic models with a Pre-trained Language Model (PLM) to maximize global and local contextual semantics. It uses a hierarchical transformer encoder and graph contrastive learning to capture global semantic information and achieve relevant sentences aligning with the gold standard summary, while minimizing redundant sentences covering sub-optimal topics. HEGEL is a hyper-graph transformer layer to capture high-order cross-sentence relationships in lengthy documents. It integrates various sentence dependencies, such as latent topics, keyword coreference, and section structure, enhancing the summarization process. The hyper-graph representation allows edges to connect to multiple vertices, enabling a comprehensive understanding of dependencies. This matrix establishes connections between sentences with common topics or keywords. Lodoss is a document segmentation and summarization method that learns robust sentence representations through summarization. It captures the document's structure and salient content, with an optimization regularizer based on determinantal point processes preventing redundancy. Lodoss is built on Longformer, using a double-layered inter-sentence transformer for operations.

Abstractive Text Summarization (ATS) represents a distinct approach aiming to generate summaries enriched with novel sentences not directly extracted from the source text. In contrast to extractive summarization, which rearranges existing sentences, ATS focuses on producing concise and coherent summaries by creating new content that encapsulates the essence of the source text. Various advanced models have been developed to address the challenges inherent in ATS. Liu et al. [126] employ the encoder-decoder framework of BERT in BERTSUMABS, enabling the generation of abstractive summaries by leveraging encoded information and creating new sentences that encapsulate the document's gist. Wang et al. [231] propose a two-step approach to enhance summarization models. In the first step, latent topics are independently extracted from the input text to capture underlying themes within the document. These extracted latent topics are then utilized in the second step to improve summarization model performance. Aralikatte et al. [11] leverage neural topic modeling with bag-of-words as input features, coupled with a transformer-based encoder-decoder architecture for generating abstractive summaries. Fu et al. [68] explore the extraction of topic distributions at both the document and paragraph levels, using these distributions as guidance in the abstractive summarization process. DimSum [257] integrates guidance from an extractive summarizer to enhance the performance of the abstractive summarizer, utilizing BART [115]. The combined loss function of the extractive and abstractive summarizers contributes to the generation of improved lay summaries from scientific documents. Recent state-of-the-art models for ATS include DYLE [138], FACTORSUM [66], PageSum [130], and HierGNN [174]. These models showcase advancements in generating abstractive summaries by incorporating innovative techniques and architectures. DYLE is a dynamic latent extraction mechanism that revolutionizes abstractive summarization by training both an extractor and a generator simultaneously. It calculates the probability of an output token based on each input snippet, while the generation probability is determined by the generator's weights and tokens. The extractor is optimized using two surrogate losses: the extractive oracle and consistency loss. FACTORSUM is a model that separates content selection from resource allocation to improve the effectiveness of abstractive summarization systems. It introduces an energy function that breaks down the summarization process into two steps: generating abstractive summary views to highlight significant information, and combining these views into a final summary while adhering to budget constraints and content guidance from advisor models like BART or BigBird. PageSum uses locality to reduce memory overhead and provide insightful summaries. It treats input documents as a collection of pages, with each page encoding independently and decoding producing local predictions and confidence scores. This emphasizes the importance of locality in text summarization. HierGNN is a neural encoder with reasoning capabilities, suitable for integration into sequence-to-sequence neural summarization models. It acquires a latent hierarchical graph, frames sentence-level

reasoning as a graph propagation problem, and uses a graph-selection attention mechanism for precise summaries.

2.2 Related Work: Personality Trait Identification

Due to the insufficiency of mental health resources relative to the prevailing demand, automated assistant tools emerge as valuable aids in diagnosing mental health issues, showcasing the potential of AI models in providing substantial support. These AI models exhibit promising capabilities as automated assistants, demonstrating superior performance in personality judgment compared to human evaluations [255]. Various studies have effectively leveraged machine learning techniques to discern personality traits in social media content [40, 219]. The identification of personality traits involves the utilization of diverse features, including demographic data and text data such as self-interpretation and content from social media. Pioneering work by Argamon et al. [12] employed support vector machines (SVMs) and statistical features from functional lexicons to identify personality traits. Farnadi et al. [63] extended this work by using SVM to detect personality traits based on features such as network size, density, and frequency of updating status. Additionally, Zhusupova [275] employed social media activity and demographic data to detect personality traits in Twitter users from Portugal.

Recent advancements have witnessed the application of various deep learning models to the task of identifying personality traits. Kalghatgi et al. [101] utilized neural networks, specifically multilayer perceptrons (MLP), in conjunction with hand-crafted features for personality trait detection. Su et al. [212] employed recurrent neural networks (RNN) and hidden Markov models (HMM) to identify personality traits from Chinese Language Inquiry and Word Count (LIWC) annotations extracted from dialogues. Tandera et al. [215] and Sun et al. [213] applied long-short-term-memory (LSTM) and convolutional neural network (CNN) to detect personality traits directly from text data extracted from Facebook posts. Liu et al. [124] devised a hierarchical structure based on Bidirectional recurrent neural network to predict personality traits from multi-lingual statements. Van de Ven et al. [221] demonstrated the accurate inference of extroversion from self-descriptions in LinkedIn profiles. Lynn et al. [133] employed message-level attention over Facebook posts for personality trait analysis. Majumder et al. [136] utilized psycholinguistic features [135] and hierarchical CNN for automatic personality detection. Gjurković et al. [74] utilized Sentence-BERT [182] on their self-created corpus. Kazameini et al. [105] applied an ensemble of SVMs over BERT embeddings, achieving superior performance on the Essays corpus [166] for Big Five trait classification. Mehta et al. [142] experimented with various combinations of BERT-based models and psycholinguistic features, achieving state-of-the-art results on different corpora. A comprehensive analysis of previous

models is presented in [143], while a review of perspectives is discussed in [211].

Despite the improvement in accuracy over time, these models confront limitations that impede their practical effectiveness. The intricate nature of textual representations, incorporating word-level dependencies across long distances and constituency representations, poses a challenge for sequential models alone to capture such information effectively. Moreover, pre-trained language model-based approaches, while achieving state-of-the-art results, are constrained by a 512-word limit for statements, presenting a hindrance in real-life applications as automated assistant tools. In light of these challenges, our investigation focuses on a model employing tree-transformers to capture word-level dependencies and phrasal information, complemented by a graph attention network (GAT) for combining sentence representations when generating the full statement representation. This approach, utilizing tree-transformers and GAT, exhibits the capability to preserve syntactical structure, overcoming limitations in word limits imposed on each sentence and the entire text, as observed in previous works by Kazameini et al. [105] and Mehta et al. [142].

Chapter 3

Metrics for Performance Evaluation

In this study, a comprehensive exploration of four distinct Natural Language Processing (NLP) applications is undertaken, spanning two diverse genres: scientific texts and psychological texts. Each application is subjected to unique assessments employing varied methodologies and metrics. This chapter serves to elucidate the performance metrics employed, offering a detailed account of their computation methods and providing insight into the evaluation process for each downstream task.

A brief presentation of each NLP application paired with its respective evaluation metrics is given in Table. 3.1.

Table 3.1: NLP applications paired with their corresponding evaluation metrics.

Task	**Performance Evaluation Metric**
Semantic Similarity Measurement	Accuracy, F-1 score, Balanced accuracy, Matthews correlation coefficient
Relation Extraction	F-1 score
Personality Trait Identification	Accuracy, F-1 Score
Text Summarization	ROUGE scores, METEOR

3.1 Categorizing Predictions

3.1.1 True Positive (TP)

This scenario arises when the model accurately predicts instances belonging to the positive class. Essentially, the model correctly identifies or classifies examples that genuinely pertain to the positive class.

3.1.2 True Negative (TN)

This situation occurs when the model accurately predicts instances belonging to the negative class. In this context, the model correctly identifies or classifies examples that do not fall within the positive class.

3.1.3 False Positive (FP)

Also acknowledged as a Type I error, this situation unfolds when the model incorrectly predicts an instance as belonging to the positive class when, in reality, it belongs to the negative class. In essence, the model erroneously signals the presence of the positive class.

3.1.4 False Negative (FN)

Also recognized as a Type II error, this scenario transpires when the model inaccurately predicts an instance as belonging to the negative class when, in fact, it belongs to the positive class. In this instance, the model neglects to identify the presence of the positive class.

3.2 Accuracy

Accuracy stands as a ubiquitous metric for assessing the overall efficacy of a classification for semantic similarity measurement model. It gauges the proportion of correctly classified instances relative to the total dataset [200, 234]. The accuracy formula is expressed as:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{3.1}$$

where, TP stands for true positive, TN for true negative, FP denotes false positive and FN is false negative. While accuracy provides a straightforward evaluation, its applicability may be limited, especially with imbalanced datasets. In scenarios where one class prevails significantly, a model may achieve high accuracy by predominantly predicting the majority class, potentially neglecting the minority class. Thus, it is prudent to complement accuracy with additional metrics like precision, recall, the F-1 score [236].

3.3 Precision

Precision [234, 236] gauges the accuracy of positive predictions made by the model. It is computed as the ratio of true positive (TP) predictions to the sum of true positives (TP) and

false positives (FP). Precision specifically focuses on the model's correctness in predicting the positive class, offering insights into the accuracy and reliability of such predictions. The formulation for precision is as follows:

$$Precision = \frac{TP}{TP + FP} \tag{3.2}$$

3.4 Recall

Recall [236] assesses the model's proficiency in capturing all pertinent instances of the positive class. Its calculation involves the ratio of true positive (TP) predictions to the sum of true positives (TP) and false negatives (FN). Recall is particularly focused on evaluating the completeness of the model's predictions for the positive class, providing insights into its ability to identify and capture all relevant instances. The formulation for recall is as follows:

$$Recall = \frac{TP}{TP + FN} \tag{3.3}$$

3.5 F-1 Score

The F-1 score, a frequently employed metric in machine learning and statistical analysis, serves as a pivotal tool for assessing the effectiveness of a classification or semantic similarity measurement model, particularly in the context of imbalanced datasets. By harmonizing precision and recall, the F-1 score encapsulates a balanced evaluation, offering a comprehensive gauge of the model's performance [236]. The formulation for the F-1 score is as follows:

$$F1\text{-}score = \frac{2 * Precision * Recall}{Precision + recall} \tag{3.4}$$

Spanning a scale from 0 to 1, the F-1 score attains higher values to denote superior model performance. Its utility shines in scenarios marked by class imbalance, where one class substantially outweighs the other. In such instances, conventional accuracy metrics may prove misleading, as a model could achieve elevated accuracy merely by predicting the majority class. The F-1 score, in contrast, offers a more equitable assessment by accounting for both false positives and false negatives, thereby providing a nuanced evaluation of the model's effectiveness.

3.6 Balanced Accuracy

For instances of imbalanced datasets, an alternative for evaluating binary classifier performance is Balanced Accuracy (BACC) [33]. Traditional accuracy metrics may falter when faced with imbalanced datasets, particularly when the model exhibits a bias toward the class with most samples. In such cases, if the model consistently predicts the majority class, the conventional accuracy metric would mirror the more frequent class's dataset proportion. This misleadingly high accuracy, however, does not reflect the model's generalization capacity.

Consider the example where class "A" comprises 90% of a dataset and the remaining 10% belongs to class "B". If a binary classifier labels all samples as "A", the traditional accuracy would erroneously be 90%. BACC addresses this limitation by incorporating the average recall obtained for both classes. It yields values within the range of (0,1) or as a percentage. A higher BACC signifies superior model performance [200].

The BACC is computed as follows:

$$BACC = \frac{\frac{TP}{TP+FN} + \frac{FN}{TN+FP}}{2} \tag{3.5}$$

In the example above, where 90% of the dataset belongs to class "A" and 10% to class "B", the Balanced Accuracy (BACC) calculation yields 0.45 (or 45%).

3.7 Matthews Correlation Coefficient

Another suitable metric when evaluating a binary classifier's performance when confronting an imbalanced dataset [168] is the Matthews Correlation Coefficient (*MCC*). The principal advantage of the Matthews Correlation Coefficient lies in its comprehensive consideration of the four prediction categories. This feature renders it particularly valuable in scenarios involving highly imbalanced datasets [30, 55, 140]. By incorporating these elements, *MCC* produces a balanced assessment of the model's performance.

The *MCC* score falls in the range -1 to +1. A score of +1 signifies perfect classification performance, while -1 indicates a complete mismatch between the model's predictions and the true observations. An *MCC* score of 0 implies a performance equivalent to random predictions [200, 235].

The *MCC* score for a binary classifier operating on an imbalanced dataset is calculated

using the following equation:

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{3.6}$$

3.8 ROUGE

ROUGE (Recall-Oriented Understudy for Gisting Evaluation) stands as a suite of metrics essential for the automated assessment of machine-generated text, particularly applied in domains like text summarization and machine translation. Its fundamental objective is to gauge the concurrence between the text produced by machines and the corresponding human-generated reference summaries. ROUGE holds a distinct recall-oriented character, emphasizing the recall of important information in the generated text compared to the reference summaries. The common practise for analysing the performance of machine summarizers is to assess unigram ROUGE (*ROUGE-1*), bigram ROUGE (*ROUGE-2*), and longest common sub-sequence ROUGE (*ROUGE-L*). ROUGE metrics span from 0 to 1, with elevated scores serving as indicators of heightened resemblance between the automatically generated summary and the reference [238].

3.8.1 ROUGE-1: Unigram ROUGE

ROUGE-1 orchestrates an assessment of the alignment of unigrams (individual words) between the machine-generated text and the human-authored reference summaries. The computation entails precision, recall, and F1-score, which are articulated as follows:

$$Recall = \frac{Number\ of\ overlapping\ unigrams}{Total\ number\ of\ unigrams\ in\ reference\ summary} \tag{3.7}$$

$$Precision = \frac{Number\ of\ overlapping\ unigrams}{Total\ number\ of\ unigrams\ in\ generated\ summary} \tag{3.8}$$

$$Rouge\text{-}1 = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{3.9}$$

3.8.2 ROUGE-2: Bigram ROUGE

ROUGE-2 is dedicated to scrutinizing the correspondence of bigrams (successive pairs of words) between the machine-generated text and the reference summaries. Its computational

essence echoes that of *ROUGE-1*:

$$Recall = \frac{Number\ of\ overlapping\ bigrams}{Total\ number\ of\ bigrams\ in\ reference\ summary} \tag{3.10}$$

$$Precision = \frac{Number\ of\ overlapping\ bigrams}{Total\ number\ of\ bigrams\ in\ generated\ summary} \tag{3.11}$$

$$Rouge\text{-}2 = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{3.12}$$

3.8.3 ROUGE-L: Longest Common Sub-sequence ROUGE

ROUGE-L, in contrast, focuses on determining the longest common subsequence (*LCS*) of words between the machine-generated text and the reference summaries. *LCS* is identified as the longest sequence of words shared between the two. *ROUGE-L* is computed as follow:

$$Recall = \frac{Length\ of\ LCS}{Total\ number\ of\ words\ in\ reference\ summary} \tag{3.13}$$

$$Precision = \frac{Length\ of\ LCS}{Total\ number\ of\ words\ in\ gnerated\ summary} \tag{3.14}$$

$$Rouge\text{-}L = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{3.15}$$

3.9 METEOR

METEOR (Metric for Evaluation of Translation with Explicit ORdering) stands as a metric widely applied for the evaluation of automatically generated summaries or translations. It intricately amalgamates precision, recall, and alignment-oriented components, presenting a holistic gauge of how effectively the generated text aligns with reference summaries. The evaluation metric is founded upon the harmonic mean of unigram precision and recall, wherein a nuanced weighting system accords greater importance to recall over precision. METEOR is intricately crafted to exhibit language-agnostic features, accommodating variations in word order, stems, and synonyms using WordNet [92, 237, 239].

Initially, it constructs an alignment between the generated summary and the reference summary which is a set of mappings between unigrams and computes precision P as follows:

$$P = \frac{m}{w_t} \tag{3.16}$$

Here, m is the number of common unigrams found between the reference and generated summaries and w_t is the number of unigrams present in the generated summary. Then unigram

recall R is computed as:

$$R = \frac{m}{w_r} \tag{3.17}$$

where, w_r is the number of unigrams present in the reference summary. This unigram precision and recall are combined to compute the harmonic mean with recall weighted 9 times more than precision following the equation below:

$$F_{mean} = \frac{10PR}{R + 9P} \tag{3.18}$$

Then a chunk penalty (p) is computed based on the number of chunks (c) in the generated summary that map to the chunks from the reference summary following:

$$p = 0.5\left(\frac{c}{u_m}\right)^3 \tag{3.19}$$

where a chunk is characterized as a grouping of single words that are contiguous in both the proposed text and the original reference. The greater the length of contiguous matches between the text being evaluated and the reference text, the fewer chunks are identified. u_m represents the unigrams in the generated summary. Finally, the METEOR score (M) is computed as:

$$M = F_{mean}(1 - p) \tag{3.20}$$

Chapter 4

Semantic Similarity Measurement

Semantic similarity measures the distance between two pieces of text in a semantic space. In scientific publications, a citation does not refer to the exact span of text that is being referred to in the referenced article. Connecting the citation to this span of text is called citation linkage. This chapter covers two of our works that have formulated the citation linkage task as a semantic similarity measurement problem.

This chapter contains two articles that deal with citation linkage as a semantic similarity task: **"Building a Synthetic Biomedical Research Article Citation Linkage Corpus"**, and **"BioCite: Citation Linkage Framework for Biomedical Research Articles"**.

Although published later than the second article, the former delineates the creation of a silver standard corpus (one that is generated and annotated through mechanical means) and its validation through rigorous analysis of a statistically representative sample of the corpus. The creation of a silver standard corpus was necessitated by the lack of a class-balanced gold standard citation linkage corpus (one that is human annotated) of a size required for deep learning that reduces the bias towards the largest class.

In the latter article, we introduce two ensemble siamese architectures tailored for this task. In the context of the siamese architecture, four sequential and four tree-structured neural networks serve as sentence encoders, with our experiments consistently revealing the superior performance of the tree-structured models. This finding underscores the importance of incorporating additional syntactic information to enhance the semantic representation of the sentences. Ensembles of the tree-structured neural networks show a further performance increase. The best encoder model for the BioCite framework concatenates the feature representations of the best performing constituent and dependency tree neural networks as the sentence representation.

4.1 Building a Synthetic Biomedical Research Article Citation Linkage Corpus

This section is based on the paper titled "Building a Synthetic Biomedical Research Article Citation Linkage Corpus" co-authored with Robert E. Mercer that appeared in *Proceedings of the Thirteenth Language Resources and Evaluation Conference (LREC 2022)* [203].

Citations are frequently used in publications to support the presented results and to demonstrate the previous discoveries while also assisting the reader in following the chronological progression of information through publications. In scientific publications, a citation refers to the referenced document, but it makes no mention of the exact span of text that is being referred to. Connecting the citation to this span of text is called *citation linkage*. In this paper, to find these citation linkages in biomedical research publications using deep learning, we provide a synthetic silver standard corpus as well as the method to build this corpus. The motivation for building this corpus is to provide a training set for deep learning models that will locate the text spans in a reference article, given a citing statement, based on semantic similarity. This corpus is composed of sentence pairs, where one sentence in each pair is the citing statement and the other one is a candidate cited statement from the referenced paper. The corpus is annotated using an unsupervised sentence embedding method. The effectiveness of this silver standard corpus for training citation linkage models is validated against a human-annotated gold standard corpus.

4.1.1 Introduction

There are a variety of formats, writing styles, and purposes for different types of written documents. It is possible for a research article to reflect a current trend in the field of study, a new invention, or a novel approach to solving a specific problem. During the process of writing a research paper, the author examines past studies that are either important in solving the topic at hand or have impacted the author's current research paper ideas. Using a *citation* is the process to refer to another article in the current research article [86]. In this way, citations serve as bridges between different research papers. Citations free up the authors' time by removing the need to repeatedly write the same thing. While doing so, it provides readers with some context for the issues being discussed in the body of the piece.

The concept of citation indexing was first introduced in 1964 by Garfield et al. [70] where indexes contain the entirety of the references in a research document. Since then, various analyses of citing have been presented (e.g., [184]). In biochemistry and physics research papers, Garzone and Mercer [71] presented a method for determining the objectives of different cita-

Table 4.1: Sample citations and the intended reference sentences that correspond (from: [86])

Example 1	Citing Statement	Formalin fixation, the most often used fixative in histology, has various advantages, including ease of tissue manipulation, optimal histological quality, long-term preservation capability, and widespread availability at a reasonable cost. [91]
	Cited Statement	The advantages of using formalin fixation are simplicity of tissue handling, the ability to store wet material for an extended period of time, and its inexpensive cost. [104]
Example 2	Citing Statement	DNA samples are frequently harmed by exposure to excessively acidic environment. [230]
	Cited Statement	DNA is fairly stable in mildly acidic solutions, although the beta glycosidic link in the purine bases is hydrolyzed at around pH4. [29]
Example 3	Citing Statement	Different PCR buffer systems and/or Taq polymerases may produce variable results in real time PCR. [91]
	Cited Statement	There is a significant disparity between the outcomes obtained using the various DNA polymerase-buffer solutions. [241]

tions. Furthermore, citation aids in the tracking of logical argumentation throughout multiple research articles [144]. Citation is commonly used to maintain the trail of scientific research argumentation across different scientific papers [162] and to summarise these documents [176].

When writing scientific research publications, citations are used when referring to a source of inspiration for a cited idea. In the case of experimental biomedical research, only a small portion of the referred material, which can be from the methodological, result, or any other sections of the cited document, is often relevant. Applications like the ones listed above would benefit from being able to extract just that relevant portion of the cited document's text. In addition, readers would not have to read an entire referenced document in order to locate the mentioned text span.

The citation linkage task for biomedical literature is a complex process: a chemical compound can be presented in multiple ways; the reactions between different drugs, chemical components, and genes can be described in very different manners; and for research articles from different sub-domains of this field, this information can be represented in different ways. Furthermore, not a lot of resources are available for deep learning this task as annotating a large corpus takes a lot of time and the annotators require domain-knowledge. At the same time, deep learning based models are data hungry and require a lot of annotated data for such task. A few corpora for the citation linkage task are currently available, but almost all are for the domain of computational linguistics research articles, not for biomedical research literature [122].

The objective of this paper is to present a method for generating a synthetic silver standard corpus for the citation linkage task for biomedical research articles and to introduce a corpus containing 74,568 sentence pairs to the research community. This corpus contains sentence

pairs that are tagged as being semantically similar or not. However, since we are using semantic similarity as a proxy for citation linkage, the corpus is intended to train models which view the citation linkage task as a textual semantic similarity measurement task in the same way as Li et al. [122]. We call this corpus a synthetic corpus as the dataset is annotated by unsupervised sentence embedding models, not by humans. And finally, the effectiveness of this dataset is assessed by testing some linear and tree-structured neural network models, which are trained with this silver corpus, on a human annotated gold corpus. The following is how the remainder of the paper is organised: The citation linkage task is discussed in Section 2 while Section 3 provides some relevant research which tackles the citation linkage task by means of assessing textual semantic relatedness between the citing and cited text spans. Data collection, data cleaning, and the automatic silver corpus creation steps are discussed in Section 4. In Section 5, the assessment of the effectiveness of this corpus is analyzed. Finally, this paper concludes with a brief summary of this work along with some directions for future research.

4.1.2 Citation Linkage

Citations create a semantic connection between the articles that are citing and the manuscripts that are being cited. While writing a research article, the authors use reference articles to support their findings and hypotheses. At the same time, they try to acknowledge the findings of the other researchers. Mentioning others' works is also important to show the significance and improvements brought by the authors with their current work.

A citation inside a research article refers to a section of the reference paper known as the *citation context* [86]. An idea or issue addressed in the referenced work is often the focus of this citation context. The citation intends to give some insight about the apposite background information to the reader so the concept of the ongoing paper becomes more understandable to them. It is possible to identify the methods, instruments, or discoveries and hypotheses in a cited publication by looking at the citation context. An author may adapt the method mentioned in the citing paper or modify it to some extent so that the performance improves or becomes compatible to the domain where he/she wants to deploy that method. Moreover, the author may conduct some experiments based upon the hypothesis of the cited paper. References to those used methods and hypotheses help the readers to easily grasp the ideas presented in the ongoing paper.

Citations, on the other hand, do not specify which part of the referenced article is being alluded to; rather, they simply state the title of the cited piece. As a result, if a reader is interested in learning more about the issue, he or she has to study the entire cited document. Readers, on the other hand, like research articles that provide them with specifics on the findings that were

made during the study with clear and specific background knowledge. This necessitates a clear understanding of the influences that have shaped this work.

A few examples of citation sentences and their related reference sentences in the cited publication are shown in Table 4.1. In Example 1 a paraphrase of the cited sentence is given which incorporates common words in a different sequence in the citing sentence. The term "pH4" is replaced by "excessively acidic environment" in the second example. It is necessary to map the pH scaling to the acidic situation to connect these two ideas. The citation sentence in Example 3 interprets the target sentence's information. It is obvious from these examples that accurate mapping between sentences and words is necessary for creating the relationship between the citing and referenced sentences.

This paper presents a synthetic silver standard corpus for training models to solve the citation linkage task for biomedical research articles by means of measuring semantic relatedness between the citing and candidate cited statements. Usually, the citation context can comprise from one single sentence to multiple paragraphs. However, models trained on this corpus can link related sentences from the cited paper given the citing sentences from the ongoing paper. This corpus comes with sentence pairs where one sentence in each pair is the citing statement and another sentence in the pair is the candidate cited statement from the reference paper. The sentence pairs in this corpus are labeled with either 0 or 1, where 1 indicates the sentences in the pair are semantically similar and 0 denotes dissimilarity.

4.1.3 Related Works

There has been a significant amount of work done to analyse citations in scientific research publications as a result of growing interest in citations [70, 71]. One approach is using citation analysis to figure out which area (such as the abstract, introduction, methodological description, result analysis and discussion of the findings) of a cited article is being referenced by a certain citation sentence. An exact citation span cannot be determined using this this type of analysis.

To help with the citation linkage task, the CL-SciSumm Shared Task is examining three different aspects: finding the text span in the referenced paper that best captures each citation sentence (a "citance"); identifying the discourse facet of each cited text span; and the reference paper's summarization using text spans referenced by several citances. The last two tasks go beyond the scope of the current paper. Text granularity considered in the first task are complete sentences, fragments of sentences, and up to five sequential sentences. In this study, while creating the corpus, we considered single sentences as the cited text span. A corpus of computational linguistics research papers is used in the CL-SciSumm Shared Task.

For the CL-SciSumm-17 shared task, Li et al. [121] used Jaccard similarity and inverse document frequency to assess which sentence pairs in citing and cited sources were linked to one another. Li et al. [119] computed the cosine similarity between sentence vectors. These sentence vectors were the concatenations of the corresponding words' 200 dimensional vectors computed from word2vec [146]. In this work, they applied a convolutional neural network over these sentence representations for generating better feature representations. Gidiotis et al. [72] fine-tuned BERT for generating sentence representations for the very same task. Umapathy et al. [220] used the Rapid Automated Keyword Extraction Algorithm [187] for detecting key-phrase similarity and a BERT-based model for detecting citation text span.

Regrettably, just a few works in the biomedical field have attempted this citation linking endeavour. And that's why only one gold standard human annotated corpus is available for this task in the biomedical field. In 2017, Houngbo and Mercer [85] created a small expert-annotated corpus consisting of sentence pairs from the biomedical area and used different traditional machine learning algorithms for textual matching operations to establish a framework for the citation linkage task.

4.1.4 Corpus Creation

In the biomedical domain, the only human annotated gold standard corpus available is from Houngbo and Mercer [85]'s work. This corpus covers texts and citations only from the methodological sections from the biomedical research articles. The citation text span in this corpus is limited to only one sentence. So, the models trained on this corpus are designed for measuring semantics of the sentence pairs, though the citation text spans in scientific research papers may cover one or multiple sentences and from different portions of the articles. The corpus is annotated by experts with proper domain knowledge and contains 3857 sentence pairs with 23 citing statements. The sentence pairs are annotated on a scale of 1 to 5 ((minimum to maximum similarity between the citing and candidate cited statement) and 0 (no similarity between citing and candidate cited sentence).

The major problem while working with this corpus is the highly imbalanced proportion between positive and negative samples. Out of these 3857 samples present in this corpus, only 81 samples are annotated with similarity score 4 and 5. That's why models trained with this corpus become highly biased towards the negative outcome. On the other hand, annotating a corpus with a sufficient number of samples which is balanced in proportion of the positive and negative samples is a very time consuming process and demands expert domain knowledge. And without such a dataset, it is tough to train data hungry deep learning models for the citation linkage task in the biomedical domain. To overcome these shortcomings, we present our

Table 4.2: Regex for detecting distinct patterns in the data

Purpose	Regex command
Capture equations	[⟨⟨{(\[\s]*([^\s⟨⟨{(\[)))}\]%*+/:<=>\-\^|~¬±·×÷ϴθ— ←→↔↦≒⇔∀∇∈∉∈∌∏∑∓∘∨∝∧∩∪∮≁≤≥≰⊂⊄⊆⊇⊈⊑⊕ ⊙⊣⊤⊥∪·|⇊▷○◦⟶⇊]+(\s*[⟨⟨{(\[]+\s*)+)?(∃|∫|log|ln)?\s* ([^\s⟨⟨{(\[)))}\]]+|(\d+[\.,])+\d+)?[)))}\]]*([[⟨⟨{(\[\s]*[%*+/: =>\^|~¬±·×÷ϴθ— ←→↔↦≒⇔∀∇∈∉∈∌∏∑∓∘∨∝∧∩∪∮≁≤≥≰⊂⊄⊆⊇⊈⊑⊕ ⊙⊣⊤⊥∪·|⇊▷○◦⟶⇊]+\s*- ?[⟨⟨{(\[\s]*(∃|∫|log|ln)?\s*([^\s⟨⟨{(\[)))}\]]+|(\d+[\.,])+\d+) \s*([⟨⟨{(\[\s]*[^\s⟨⟨{(\[)))}\]%*+/:<=>\^|~¬±·×÷ϴθ— ←→↔↦≒⇔∀∇∈∉∈∌∏∑∓∘∨∝∧∩∪∮≁≤≥≰⊂⊄⊆⊇⊈⊑⊕ ⊙⊣⊤⊥∪·|⇊▷○◦⟶⇊]+|(\d+[\.,])+\d+|-)?[\s)))}\]]*)+
Capture numbers with no prior symbol	(^\d+| \K\d+)((\.\d+)|\^\d+|e[\-+]?\d+(\.\d+)?)?(?![\-\w])
Capture numbers with prior symbol	(?<!(\w|\d))[\-+]\d+((\.\d+)|\^\d+|e[\-+]?\d+(\.\d+)?)?
Capture citations	\[\d+([,\-]\d+)*\]
Capture chemical names ending with ∝ and β	-\s*∝|-\s* β

synthetic corpus of 74,568 sentence pairs from 2,736 citing and 138 cited papers covering 3 biomedical sub-domains: chemical biology, biochemistry and cell biology. We call this corpus synthetic as no human supervision is used for data annotation. Rather, the unsupervised sentence embedding model Sent2Vec [161] is used to serve this purpose. For assessing the effectiveness of this synthetic corpus, models are trained with this corpus, but validated and tested against the gold standard corpus of Houngbo and Mercer [85]'s work. However, the scoring factor of this gold standard corpus is modified for our work. Similarity scores of the samples with score 0 to 3 are replaced by 0 and samples annotated with similarity score 4 and 5 are labelled with 1. We chose Sent2Vec for creating the silver standard synthetic corpus because of it's ability to work with out of vocabulary words and doesn't require any pre-trained word embeddings [161].

The overall corpus creation process is described here in three steps: i) data collection, ii) data cleaning, and iii) data annotation.

4.1.4.1 Data Collection

Sent2Vec, like all other unsupervised models, demands a large amount of training data. That's why for training the model, 4,843,756 sentences from 28,310 research documents are accumulated. These documents from more than 90 different fields of biomedicine are extracted from BioMed Central.

For the purpose of creating the sentence pairs with citing and cited sentences, 138 articles from the fields of cell biology, biochemistry, and chemical biology were chosen at random from a pool of these 28,310 research papers and these papers are considered as the cited reference papers. A total of 2,736 citing papers (cite at least one of the papers from these 138 reference papers) are collected manually and from them, only the relevant citing statements are extracted.

4.1.4.2 Data Cleaning

As the research articles are accumulated from different biomedical sub-domains, they come with a variety of writing formats and representations. Furthermore, the same equations may be represented in different ways with different symbols and variable names. That's why, to avoid confusion, all of the equations in these cited and citing papers are replaced with "$< equ >$". All of the isolated numbers are also replaced by "$< num >$". However, if any number comes as a part of any chemical compound name, it is preserved without any modification. The documents contain some symbols which have no importance in terms of representing the semantics neither at the sentence level nor at the document level. Such symbols are identified and deleted from the data. Citation indexes like "[xx]" are also deleted as they have no semantic value. Some Greek letters have different usages in different scenarios. For instance, α has no importance in terms of semantics when it is used as a variable in an equation, but, when it comes as a part of a chemical name, like "α-carbon", it differentiates the chemical from other variants. That's why when such Greek letters appear as a part of equations, they are kept untouched and the whole equation is replaced with "$< equ >$". But, when these Greek letters come as a part of chemical names, they are replaced with their written form (e.g., α in "α-carbon" is replaced by alpha). Finally, symbols which are represented in multiple ways are replaced with their corresponding common format of representations and then all the data are lower-cased. Table 4.2 shows the regex commands used for the data cleaning step. Finally, all the unnecessary symbols are deleted.

4.1.4.3 Data Annotation

Following the cleaning of the sentences, the unsupervised sentence embedding model Sent2Vec is trained. This step is necessary in order to properly annotate the pairs of citing and candidate

Table 4.3: Hyper-parameter settings used for training Sent2Vec. The selected parameter values are marked as bold.

Hyper-parameters	Values
Embedding Dimension	700/600/**500**/400/300/200
Iterations	20/15/**10**/5
Window Size	**20**/10
Learning Rate	**0.2**/0.1/0.05/0.01
Negative Samples	**10**
Loss Function	softmax/ Hierarchical softmax/ **Negative sampling**
Sampling Threshold	**0.0001**

cited sentences. That is why Sent2Vec is trained on the data using a variety of parameter settings. Table 4.3 illustrates different hyper-parameter settings. Hyper-parameter values for the optimal sentence embedding are indicated, as well.

In order to produce a sentence pair for each sentence in the cited article, after the data has been cleaned, sentence pairs are formed in which one sentence is taken from the cited article and the other is taken from the citation. A total of 522,398 sentence pairings are generated in this step.

The Sent2Vec model is then used to generate the vector representations of individual sentences from each pair and after that, cosine similarity between sentence vectors for citing and candidate cited statements in each pair is computed. Performance is evaluated against the gold standard validation set from Houngbo and Mercer [85]'s work for varied cutoff cosine similarity values. This validation set consists of 800 sentence pairs with 20 randomly chosen positive samples. Samples with cosine similarity score more than the cutoff are tagged with similarity score 1 (indicates the citing and the candidate cited statements are semantically similar) and 0 otherwise (there is no similarity between the citing and the candidate cited sentences). This cutoff value is determined by looking at the Balanced Accuracy, Matthews correlation coefficient (MCC), and F1 score metrics over the validation dataset. Sentence vectors with 500-dimensional representations and a cutoff value of 0.57 produce the best results.

However, after this approach it is found that the vast majority of these 522,398 sentence pairs have annotation value 0. Any model will be biased towards the negative outcome, if it is trained with this corpus. Because of this, 74,568 samples are selected from these pairs to ensure that the positive and negative samples are evenly distributed. For this selection process, all the positive samples (annotated with similarity value 1) are retained, while for each citing statement, n negative samples are chosen randomly where for that citing sentence n positive samples are found. Thus, this evenly distributed silver standard corpus with 74,568 is gener-

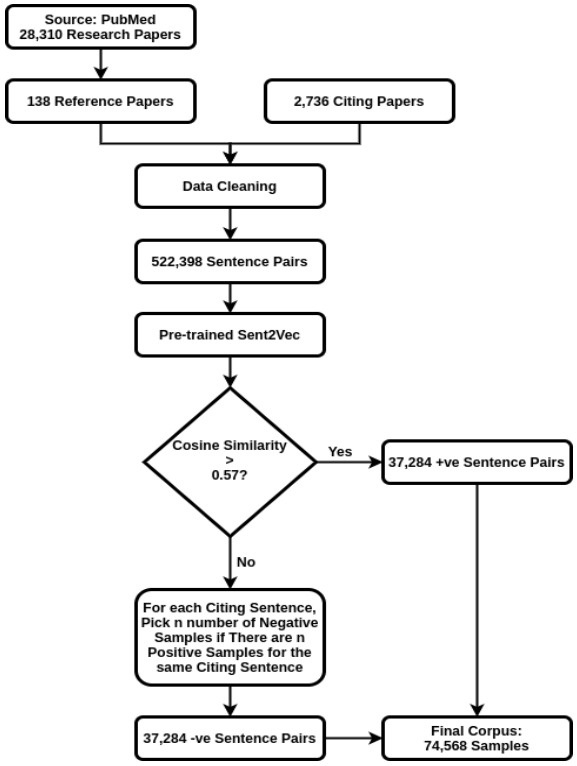

Figure 4.1: Annotated sentence pair creation for synthetic corpus build-up.

ated. The whole corpus creation process is portrayed in Fig 4.1.

4.1.5 Evaluation of the Synthetic Corpus's Effectiveness

We have evaluated the quality of the synthetic corpus in two steps. In the first step, an analysis is performed on a statistically valid sample of the corpus (95% confidence, 3% margin of error) with some human annotators' help, and in the second step, various sequential and tree-structured models are trained with this corpus and the trained models' performances are evaluated on a gold standard test set. For the statistical analysis, from the pool of 74,568 citing and candidate-cited sentence pairs, we randomly selected 750 positive and 750 negative samples for evaluation of the annotation quality (labelled accordingly in the synthetic corpus). Two groups of expert annotators both annotated the 1500 pairs of sentences. There were three people in each group, and they each annotated 500 samples. In other words, each 500-sample chunk was annotated by two people, one from each group. Each reviewer also expressed their level of confidence in the sample annotations they were given. The inter-annotator reliability between the human experts and between the human experts and the synthetic corpus was then calculated using Cohen's κ. One group found 731 positive and 769 negative examples in 1500 sentence pairings, while the other found 709 positive and 791 negative. The annotator groups

agreed upon 706 positive samples and 765 negative samples. This study's κ reliability factor is 0.96. For 715 and 701 positive samples, the synthetic silver corpus and the first and second annotator groups agreed on annotation decisions, respectively. In both situations, the annotators agreed with the synthetic silver corpus on all of the negative samples' annotations. In terms of κ, the first group of annotators and the mechanically created corpus have an inter-rater reliability of 0.95 and between the second annotator group and the synthetic corpus, 0.93. By comparing these two sets of results, we can see that the automatic annotations closely match the expert annotations. When evaluating these high κ values, it is important to keep in mind that the annotators were given a 50/50 distribution of positive and negative samples .

For assessing the effectiveness of the introduced silver standard synthetic corpus, we conducted three experiments. In the first experiment, we trained different sequential and tree-structured deep neural network models with 3057 samples with 61 positive samples from the gold standard dataset [85] and tested them against 400 sentence pairs containing 10 positive sentence pairs from the same dataset. The remaining data from this dataset was used for the validation purpose. In the second experiment, we trained the same models with the synthetic silver standard data and then validated, and tested against the gold standard data just like we did in the first experiment. If the results are found better in the second case, then it proves the effectiveness of training models with the proposed synthetic corpus. In our last experiment, 3057 samples containing 61 positive samples are used for the testing purpose and the remaining data are used for the validation of the models. Results from this experiment shows how good the models perform on a larger portion of the gold standard dataset if they are trained with our synthetic dataset.

The base for all of the models used for the assessment of the quality of the synthetic corpus is the Infersent [53] architecture. As the sentence encoders in the Infersent architecture, four sequential and two tree-based models are used. The basic working principle of Infersent is the use of siamese sentence encoders and applying concatenation, absolute difference, and point wise multiplications over the sentence representations computed from the identical encoders. Finally, this feature representation is used for the downstream tasks. In our experiments, one encoder is fed with the citing sentence and the other encoder is fed with the candidate cited sentence. Then after the encoding and the above stated three operations are done, it is fed to a two-way *softmax* classifier layer for computing the binary semantic relatedness value. Outcome 1 indicates that the citing sentence is actually referring to the candidate cited sentence and 0 indicates the opposite.

In the Infersent architecture, Bi-LSTM with max-pooling, hierarchical CNN [269], Bi-LSTM with inner [127] and hierarchical attention [249] mechanisms, and two variants of tree-transformers, dependency (DT-Transformer) and constituency (CT-Transformer) tree-

Table 4.4: Performance analysis of different models trained with the gold corpus [85]. The test set contains 400 samples from [85]. The performance metrics are TP: true positive; FP: false positive; TN: true negative; FN: false negative, P: precision, R: recall, F1: F1 score, MCC: Matthews correlation coefficient; Acc: accuracy, BAcc: balanced accuracy.

Model	TP	FP	TN	FN	P	R	F1	MCC	Acc (in %)	BAcc (in %)
hCNN	2	0	390	8	1	0.2	0.33	0.44	98	60
Bi-LSTM & Max-Pooling	1	0	390	9	1	0.1	0.18	0.31	97.75	55
Bi-LSTM & Inner Attention	1	2	398	9	0.33	0.1	0.15	0.17	97.25	54.74
Bi-LSTM & Hierarchical Attention	1	2	398	9	0.33	0.1	0.15	0.17	97.25	54.74
CT-Transformer	2	1	389	8	0.67	0.2	0.31	0.36	97.75	59.87
DT-Transformer	1	2	398	9	0.33	0.1	0.15	0.17	97.25	54.74

Table 4.5: Performance analysis of different models trained with the synthetic silver corpus. The test set contains 400 samples from [85]. The performance metrics are the same as for Table 4.4.

Model	TP	FP	TN	FN	P	R	F1	MCC	Acc (in %)	BAcc (in %)
hCNN	7	9	381	3	0.44	0.7	0.54	0.54	97	83.85
Bi-LSTM & Max-Pooling	7	7	383	3	0.5	0.7	0.58	0.58	97.5	84.10
Bi-LSTM & Inner Attention	8	6	384	2	0.57	0.8	0.67	0.67	98	89.23
Bi-LSTM & Hierarchical Attention	8	5	385	2	0.62	0.8	0.69	0.69	98.25	89.35
CT-Transformer	9	5	385	1	0.64	0.9	0.75	0.75	98.5	94.36
DT-Transformer	9	3	387	1	0.75	0.9	0.82	0.82	99	94.62

transformers [6], are used as the encoders. All of the encoder architectures are fed with word embeddings from Bio-RoBERTa [116]. The hidden layer in all models contains 512 neurons in all cases and a stochastic gradient descent optimizer is used. The hierarchical CNN (hCNN) concatenates features from 4 layers of convolution operations and both the inner and hierarchical attention mechanisms come with 4 heads for focusing on 4 different portions of the sentences which are concatenated in the end. Both tree-transformers use 6 parallel heads with 50-dimensional key, query and value matrices and the Adagrad optimizer is used. For all of the sentence encoder models, the learning rate is initialized to 0.1. This learning rate is divided by 5 if the validation accuracy reduces in the subsequent epoch.

Tables 4.4 and 4.5 show the performances of the models over the same test set containing 400 sentence pairs from Houngbo and Mercer [85]'s human annotated corpus averaged with four similarly sized randomly chosen subsets. When the models are trained with training set data from the gold standard corpus (3057 samples containing 61 positive samples), no model could retrieve more than 2 out of 10 positive samples from the test set (Table 4.4). The overall accuracy found for all the models are always more than 97% as the data contains more than 97% negative samples. It proves that when the models are trained with this human annotated

Table 4.6: Performance analysis of different models trained with the silver standard synthetic corpus. The test set contains 3057 sentence pairs from [85]. The performance metrics are the same as for Table 4.4.

Model	TP	FP	TN	FN	P	R	F1	MCC	Acc (in %)	BAcc (in %)
hCNN	46	576	2420	15	0.07	0.75	0.13	0.20	80.69	78.09
Bi-LSTM & Max-Pooling	53	359	2637	8	0.13	0.87	0.22	0.31	88.02	87.45
Bi-LSTM & Inner Attention	54	349	2647	7	0.13	0.89	0.23	0.32	88.38	88.43
Bi-LSTM & Hierarchical Attention	56	339	2657	5	0.14	0.92	0.25	0.34	88.75	90.24
CT-Transformer	57	315	2681	4	0.15	0.93	0.26	0.35	89.56	91.46
DT-Transformer	57	301	2695	4	0.16	0.93	0.27	0.36	90.02	91.70

corpus, they are biased towards the negative outcome. But, when the same models are trained with the proposed silver standard corpus, the models retrieve 7 to 9 positive samples out of 10 correctly. The best result is found for the DT-Transformer model. It accurately determines 9 positive samples with a balanced accuracy of 94.62%. These results prove the effectiveness of the proposed silver standard dataset.

Table 4.6 shows the performance of the various models on the original gold standard training set [85] averaged with four similarly sized randomly chosen subsets when trained with the synthetic silver standard corpus. When the models are trained with the silver corpus, models achieve up to 91.70% balanced accuracy. These models utilize recent deep learning techniques and attention mechanisms which allow them to put more focus on the important portions of the text. The tree-transformer models outperform all the sequential models as they incorporate word level dependency and phrase level information. With these tree structured transformer models, 57 out of 61 positive pairs are extracted accurately. These results reflect that if the models are trained with the proposed synthetic corpus, they perform very well over the gold standard dataset.

4.1.6 Conclusion

In this paper, we introduce a synthetic silver standard corpus for the citation linkage task in the biomedical domain and also a method to annotate such a corpus without any human help or expert opinion. Performance of the models trained with this dataset reflects the effectiveness of this corpus. This corpus will be made publicly available. As we started this project a couple of years ago, we used Sent2Vec for the sentence embedding. In future work, different BERT-based models can be utilized. One limitation of this work is that the considered citation text span is limited to a single sentence only. However, in real application scenarios, the referenced text may span over multiple sentences. Keeping this in mind, we are trying to build a gold and a silver standard corpus for the citation linkage task where the text span can be single to

multiple sentences.

4.2 BioCite: A Deep Learning-based Citation Linkage Framework for Biomedical Research Articles

This section is based on the paper titled "BioCite: A Deep Learning-based Citation Linkage Framework for Biomedical Research Articles" co-authored with Robert E. Mercer that appeared in the *Proceedings of the 21st Workshop on Biomedical Language Processing (BioNLP 2022)* [202].

Research papers reflect scientific advances. Citations are widely used in research publications to support the new findings and show their benefits, while also regulating the information flow to make the contents clearer for the audience. A citation in a research article refers to the information's source, but not the specific text span from that source article. In biomedical research articles, this task is challenging as the same chemical or biological component can be represented in multiple ways in different papers from various domains. This paper suggests a mechanism for linking citing sentences in a publication with cited sentences in referenced sources. The framework presented here pairs the citing sentence with all of the sentences in the reference text, and then tries to retrieve the semantically equivalent pairs. These semantically related sentences from the reference paper are chosen as the cited statements. This effort involves designing a citation linkage framework utilizing sequential and tree-structured siamese deep learning models. This paper also provides a method to create an automatically generated corpus for such a task.

4.2.1 Introduction

Research articles from different domains use varying writing styles and formats. They serve different purposes as well. A research publication may discuss current research trends, a novel discovery, or alternative approaches to solving a problem in a given domain. While writing a research article, the author mentions prior research that was either significant in resolving the same topic or impacted the author's views mentioned in the current research paper. This referencing another document in a research piece is referred to as a *citation* [86]. This way, citations establish connections between distinct research literature as well as alleviating authors' writing burden by preventing them from having to write the same thing mentioned in another research article again. Simultaneously, it assists readers in acquiring prior knowledge about a subject that may be necessary to comprehend the ideas contained in the ongoing research work.

The idea of citation indexing was first introduced in 1964 where indexes contain the references in a research document. Citation-based bibliometrics are utilized to evaluate the significance of a research work [70]. In response to the growing popularity of citation indexing, a more critical analysis of citing was later suggested. Garzone and Mercer [71] devised a mechanism for determining the objective of a reference in biochemistry and physics research publications. Moreover, citations help to keep track of the logical argumentation across various research articles [144]. Prominent applications of citation incorporate maintaining the trail of scientific research argumentation across different research articles [162] and summarization of these documents [176].

In scientific research publications, a citation refers to the source article from which the cited notion is drawn. However, in experimental biomedical research articles, a citing sentence usually only relates to a small text span of the cited document's contents. This small span of text can be from the method section, result analysis section or any other section of the reference document [208]. The above-mentioned applications would substantially benefit if such a text span could be extracted from the original document. It would also free up the readers from having to read the full document to locate the cited piece of text.

The citation linkage task is more complicated for biomedical research papers as the same chemical or biological component has various representation formats and the use of these variations is very common in such research articles. For example, the chemical compound carbon dioxide can be represented as CO_2 as well as $O{=}C{=}O$, whereas in some articles the writers write the whole name in plain text (*carbon dioxide*). Similarly, there are multiple representations to indicate the same reactions between various genes, chemicals, and drugs. On top of that, the only human annotated corpus available for the citation linkage task in the biomedical domain is from [85] which comes with 3857 sentence pairs which are highly imbalanced with only 2% positive samples and 98% negative samples. The size and imbalanced nature of this corpus makes it difficult to train deep learning models on this dataset. To overcome this, we propose an automatically generated corpus for this task containing 74,568 sentence pairs.

This paper has two objectives: first, introducing an automatically generated corpus for the citation linkage task for biomedical research papers and second, providing a framework for this task to retrieve the cited text span from the reference paper given the citing sentence by means of measuring the semantic similarities between the citing sentence and candidate cited sentences from the referenced paper. The cited text span can be a single sentence, part of a sentence or even one or more paragraphs [85]. However, for this task this text span is restricted to a single sentence like Li et al. [121]. Considering the first objective, we introduce an automatically generated corpus containing 74,568 sentence pairs and also an approach to annotate data automatically without any human effort. The quality of the data annotation is evaluated by

annotating a portion of the dataset by human experts and then measuring Cohen's κ among the human annotators' decisions and the automatically generated annotation labels. Sentence pairs from this dataset are used only for training the models for the citation linkage task. And for the second aspect, we have investigated multiple sequential and tree-structured neural networks and presented one ensemble architecture, which we call BioCite, that computes the semantic similarity between the citing statement and all of the sentences in the referred document. The performance of the model is tested against the expert annotated dataset from Houngbo and Mercer [85] which contains citing sentences that refer to methods statements in the cited documents. The outline for the paper is: Section 4.2.2 gives a brief description of the citation linkage task and Section 4.2.3 mentions and discusses a few prominent works for the citation linkage task. Section 4.2.4 discusses the automatically generated corpus creation and the framework design. The performance of the models are reported and analyzed in Section 4.2.5. The parameters of the models are also described in this section. The paper ends with a brief summary and possible future directions of this research.

4.2.2 Citation Linkage

Citations construct semantic bridges between citing and cited manuscripts. To support the findings, claims and hypotheses, authors cite several resources while preparing manuscripts. They also try to address the results and findings of the other research works. It is also important to mention others' works, in order to demonstrate the authors' significance and progress with their current work.

A citation in any research paper focuses on some specific sections of the referenced article acknowledged as the *citation context*. This citation context often focuses on a specific idea or issue in the referenced manuscript [86]. The intent of a using citation is to provide the readers with the apposite background information for a better understanding of the concepts introduced in the citing paper. The citation context can reveal information about a cited publication's hypotheses, findings, methodologies, etc. In order to improve the performance or make the method compatible with the domain for which it is intended to be used, an author may adapt or modify the method described in the citing paper to the extent necessary. Aside from that, the author may undertake experiments based on the idea presented in a cited paper to confirm or refute the idea presented in that work. References to the hypotheses and methodologies that were employed in the referenced paper aid the readers to grasp the concepts presented in the current work.

However, citations only provide the source of information which is being referred. The current citation indexing approach does not provide a way to indicate which text span from

the cited research manuscript is actually being touched on. It provides no method other than going through the whole referenced article for the reader if he or she wants to grasp the idea properly. On the other hand, research articles that include detailed information on the study's discoveries, as well as relevant background information, are more appealing to readers. This necessity has influenced the work we are presenting in this paper.

The author can cite a paper by paraphrasing the statements from the cited paper. He or she can also elaborate some statements from the cited paper. For example in the citing statement, "DNA samples are frequently harmed by exposure to excessively acidic environment", Wang et al. [230] explains that "pH4" is an "excessively acidic environment" when citing "DNA is fairly stable in mildly acidic solutions, although the beta glycosidic link in the purine bases is hydrolyzed at around pH4." [29]. Sometimes these citations are the interpretations of the cited statements, e.g., the citing sentence "Different PCR buffer systems and/or Taq polymerases may produce variable results in real time PCR." [91] is nothing but an interpretation of the cited sentence "There is a significant disparity between the outcomes obtained using the various DNA polymerase-buffer solutions." [241]. As these examples demonstrate, precise mapping between words and sentences is required to establish a connection between the citing and cited sentences.

This paper provides a citation linkage framework for biomedical research articles along with an automatically generated corpus comprising 74,568 sentence pairs. The framework at first generates sentence pairs with the citing sentence and all the sentences from the referenced paper. Then, the model measures the semantic similarity scores between the sentences in each pair. Based on these similarity scores, it retrieves the actual cited sentences from the referenced manuscript. We have formulated this semantic similarity measurement task as a binary classification task where each sentence pair is predicted with either label 1 or label 0. Sentence pairs predicted with label 1 are selected as the cited sentences given the particular citing sentence.

4.2.3 Related Work

The study of citations in scientific research has led to a lot of work. Citation analysis attempts to identify which section (i.e., abstract of the paper, introduction of the problem statement, description of methods, analysis of result, etc.) of the referenced article this sentence refers to [70, 71]. However, this form of study cannot pinpoint the citation span.

Another type of work is to determine the citation span. PolyU [39] applied RankSVM over chunks of sentences to predict the cited text span. Baruah et al. [19] computed cosine similarity of word embeddings for the citation linkage task. Yeh et al. [253] applied majority voting to

six machine learning classifiers over the lexical, knowledge-based, corpus-based, syntactic and surface features for this task.

The CL-SciSumm Shared Task tries to solve three aspects: find the cited text span given the citation sentence ("citance"), identifying the discourse facet of the cited sentence and summarise the referred article using only the text spans that are quoted many times in the referenced document. However, the later two sub-tasks are out of the scope of this work. Ma et al. [134] applied different classifiers and voting mechanism over similarity, rule and position-based features to determine the similarity between the citing and cited statements for CL-SciSumm-17. The citation linkage between citing and cited sentence pairs was determined by Li et al. [121] utilizing inverse document frequency and Jaccard similarity. In their following works, they computed the sentence vectors by concatenating 200 dimensional word vectors [119] and then applying a convolutional neural network (CNN) over that concatenated vector representation [122]. In both cases, the cited text span is determined by measuring the cosine similarities between the citing and candidate cited statements. Other works, such as AbuRa'ed et al. [2] have also worked with the CL-SciSumm corpus.

Recently, BERT-based models have been deployed for the citation linkage task and are being used in many experiments. Gidiotis et al. [72] fine-tuned BERT to determine the referred cited sentences from the cited document. Zerva et al. [259] applied a CNN over SciBERT-based features [21] to determine which text span in the cited article is actually being referred. They concatenated the features from the BERT-based model for feature generation. Umapathy et al. [220] utilized key-phrase similarity using the Rapid Automated Keyword Extraction Algorithm [187] and a BERT-based architecture for cited text span identification.

However, only a few citation linkage works are found for biomedical research papers. Citation linkage for biomedical research articles is more challenging due to various representations of the same component. One notable work for this domain is from 2017, where Houngbo and Mercer [85] used traditional machine learning approach over their own small expert-annotated corpus. And so far, this is the only human annotated corpus for the citation linkage task in the biomedical domain.

4.2.4 BioCite: Description of the Framework

The development of the framework involves two major steps: creating a balanced automatically generated training corpus of reasonable size and building a framework for determining the referred statements from the cited document for a particular citing statement.

4.2.4.1 Corpus Creation

The only expert-annotated corpus for the biomedical domain to serve the purpose of our work is from Houngbo and Mercer [85] which comes with only 3857 sentence pairs. For training, the major problem with this dataset is the class imbalance: only 81 positive pairs which is only 2% of the corpus. Eventually, training any model with this corpus would make it biased towards negative outcome. At the same time, manually annotating enough data from biomedical and biochemical research articles for this task is time consuming. So, we have created an automatically generated corpus of 74,568 sentence pairs spanning three biomedical sub-domains: biochemistry, cell biology and chemical biology. We are calling this corpus automatically generated as no human annotation has been used for generating these sentence pairs. For the validation and testing of the models, we have used the validation and testing sets from the Houngbo and Mercer [85] corpus (800 samples with 20 positive ones for validation and 3057 samples containing 61 positives for test set). The sentence pairs in the training set are annotated with 0 (not semantically similar) or 1 (semantically similar) to make it compatible with the validation and test set.

We collected 28,310 research documents from BioMed Central spanning multiple biomedical sub-domains. From these documents, 138 are randomly chosen from the above-mentioned three sub-domains and then corresponding citing statements from 2736 papers (manually collected) citing these 138 articles are extracted manually. The citing statements are then paired-up with all of the sentences from the corresponding cited documents, ending-up with 522,398 pairs.

Sentences of each pair are fed individually to the Sent2Vec [161] model, which is trained over all of the research documents we accumulated, and the cosine similarity between the paired sentences is measured. Pairs with cosine similarity value greater than a cutoff value 0.57 (selected after testing against the validation set) are labelled 1, 0 otherwise. We experimented with different cut-off values and plotted the results on AUC and ROC curves while testing on the validation set from the expert annotated corpus [85]. From there, we chose the cut-off value for which the best validation accuracy was found. From there As there are many fewer positive samples than negative ones, for each citing statement, negative samples are randomly chosen for each citing sentence to balance the classes. In this automatically generated corpus, for each citing sentence, an equal number of positive and negative samples are preserved. The overall process of this corpus creation is illustrated in Figure 4.2.

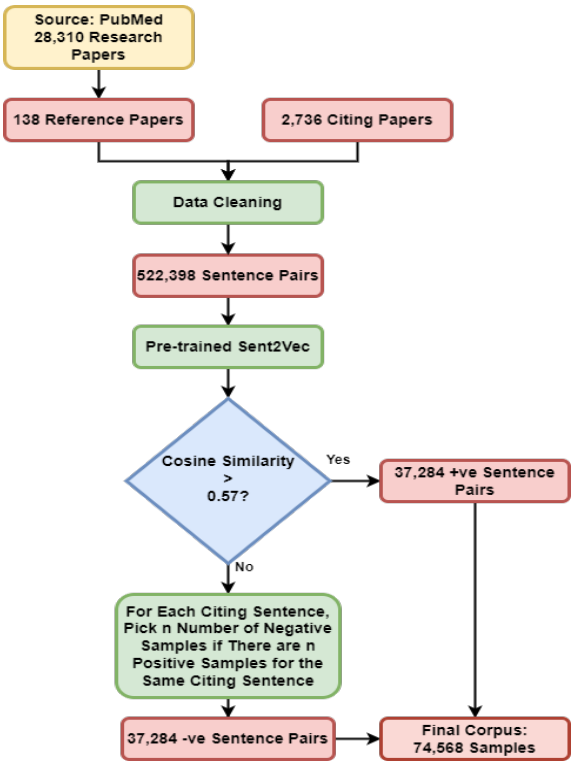

Figure 4.2: Automatically generated corpus build-up: Sentence pair creation and annotation.

4.2.4.2 Semantic Similarity Measurement Module

The aim of building this citation linkage framework is to link the citing sentence to the referenced text span in the referenced biomedical research article. To solve this challenge, we have used a variety of supervised deep learning-based models to estimate the semantic similarity between the citing and cited text span where the text span is limited to a single sentence. The predictions of these models are set to binary class labels: 0 and 1. Here 1 indicates that the candidate cited and the particular citing statement are semantically similar and it can be interpreted as the candidate cited sentence is truly being referenced by the citing sentence and if the prediction value is 0, it represents the candidate cited sentence is not being referred.

The base of the sequential and tree-structured neural network models is InferSent [53]: a siamese architecture. This is a supervised sentence representation model which is able to work with sentence pairs and has been used in many cases for semantic relatedness measurement tasks [6, 182]. The overview of the training process of InferSent for the semantic similarity measurement task is portrayed in Figure 4.3. In InferSent two identical encoder neural network topologies are used with identical parameter settings. The citing sentence (S_{citing}) and the cited sentence (S_{cited}) are encoded by them in parallel. This is followed by generating a feature map that concatenates concatenation, absolute point-wise difference, and point-wise multiplication.

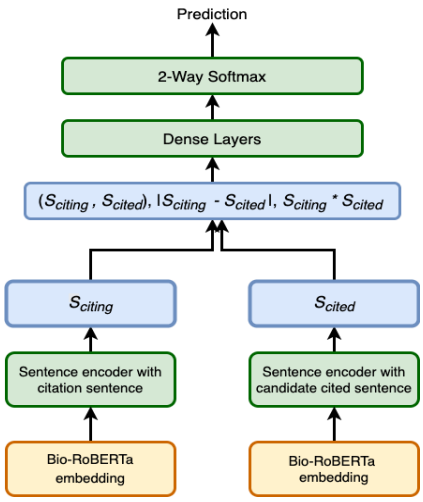

Figure 4.3: InferSent training for the citation linkage task.

This feature map is then loaded into the dense and *softmax* layers in sequence to predict the binary class label. As the encoder models, four sequential and four tree-structured neural networks are used. The functioning principles of these models are first outlined, and then the ensembles of them are discussed. The best encoder model for the BioCite framework is chosen in the end based on the performances of the investigated models.

4.2.4.3 Sequential Encoders

As the encoder for the InferSent model, four sequential models are applied. The first one is the Bi-LSTM with a following max-pooling layer. The second encoder model applies inner attention [127] over the Bi-LSTM output features for producing the sentence representations. The third encoder model utilizes the hierarchical attention [249] in place of inner attention over the Bi-LSTM. This attention mechanism was introduced for document classification where at the first layer it attends on the words for generating sentence representation and in the second layer it attends over the sentences for paragraph or document representation. As our work is limited to single sentences, we have used only the first layer of this attention mechanism. This approach is designed in such a way that it can focus on four different parts of the sentence. Thus it generates four sentence representations, which are concatenated to form the sentence vector. The last sequential encoder we investigated is the hierarchical CNN with four layers of convolution operations, each followed by one max-pooling operation. These four feature maps are concatenated in the end to generate the sentence representation vector.

4.2.4.4 Tree-Structured Encoders

Sequential neural networks provide reasonable sentence representations. However, they can't preserve structural information and miss semantic compositionality. Tree-structured neural networks, on the other hand, can preserve both semantic and syntactic properties of the text by working with the parse tree. For the tree-structured neural network models we investigated the dependency and constituency tree-transformers with both multi-head and multi-branch attention mechanisms over child nodes' representations [6]. For completeness, we provide details of these tree-transformers that are developed therein.

A constituency tree contains words at leaf nodes only, whereas a dependency tree has a word at each node. So, while traversing a dependency tree, it is required to consider both the child and corresponding parent nodes whereas for constituency tree, only after traversing every sub-tree the non-terminal intermediate nodes can be calculated. So, in both cases, the children nodes are considered. This approach [6] uses self attention mechanism for attending the child nodes. This attention mechanism uses three matrices: *key*, *value* and *query* like the transformer model [223] (Equ. 4.1).

$$\alpha = \texttt{softmax}(\frac{query\ key^T}{\sqrt{d_k}})value \tag{4.1}$$

Here d_k is the dimension of the *key*, *value* and *query* matrices. For this experiment the dimension of all these matrices are kept the same. n copies of these matrices are generated for n branches of the multi-branch attention mechanism. Here, n is the number of branches to be used. Then scaled dot product is used as in Equ. 4.2:

$$\beta_i = \alpha_{i\in[1,n]}(query_i\ \omega_i^q, key_i\ \omega_i^k, value_i\ \omega_i^v) \tag{4.2}$$

where ω_i^q, ω_i^k, ω_i^v are the hyper-parameter weight matrices for *query*, *key*, and *value*, respectively.

Following this scaled dot product operation, a residual connection is employed over these tensors β. A layer-wise batch normalization is used in the following step which is multiplied with a scaling factor τ (Equ. 4.3). Over every $\tilde{\beta}$, position-wise CNN (PCNN) is then employed (Equ. 4.4). By applying weighted summation then, the attention encoded semantic sub-spaces' representation are generated (Equ. 4.5). Here $\gamma \in \mathcal{R}^n$ is a hyper-parameter. In the end, another residual connection is established with `BranchAttn` which is then fed to a non-linearity function tanh and an element-wise summation function `EWS` is done to produce the parent node

representation (Equ. 4.6) [6].

$$\tilde{\beta}_i = \texttt{LayerNorm}(\beta_i \omega_i^b + \beta_i) \times \tau_i \tag{4.3}$$

$$\texttt{PCNN}(x) = \texttt{Conv}(Relu(\texttt{Conv}(x) + b_1)) + b_2 \tag{4.4}$$

$$\texttt{BranchAttn} = \sum_{i=1}^{n} \gamma_i \texttt{PCNN}(\tilde{\beta}_i) \tag{4.5}$$

$$\texttt{ParentNodeRep} = \texttt{EWS}(\tanh((\tilde{\chi} + \chi)\omega + b)) \tag{4.6}$$

For multi-head attentions, attention matrices *key*, *value* and *query* are projected h times [223] and it is calculated as follows:

$$\texttt{MultiHead}(query, key, value) = \texttt{Concat}(\texttt{head}_1, ..., \texttt{head}_h)W^O \tag{4.7}$$

where, for each head,

$$\texttt{head}_i = \alpha(queryW_i^q, keyW_i^k, valueW_i^v) \tag{4.8}$$

All of the Ws are the hyper-parameter matrices which get updated during training.

4.2.4.5 Ensemble Architectures

After investigating the sequential and tree-structured neural network models, we experimented with two ensemble models. The first ensemble architecture utilizes all the models investigated here. After all the models are trained separately, each sentence pair is fed to all the models in parallel. Each model individually predicts the semantic similarity score and in the end, the final similarity value is selected by applying a winner-takes-all approach [188] over all the predictions. In the second approach we used only the tree-transformer models. The dependency tree-transformer is able to preserve the word level dependency between different part of the sentence, whereas the constituency tree-transformer can preserve phrase-level information. To benefit from both of these models, we concatenated the feature representations generated from both of the tree-transformers and used it as the vector representation of the sentence. This sentence vector is then fed to a multi-layer perceptron for the similarity score prediction.

4.2.5 Experimental Setup and Result Analysis

In this section, the experimental setup and the results of the models investigated for the citation linkage task are discussed. As the human annotated test data is highly imbalanced, apart from

Table 4.7: Statistics of the annotations by the experts and the automatically generated corpus for the 1500 samples

	Annotator Group 1	Annotator Group 2	The Automatically Generated Corpus
Positive samples (in total)	731	709	750
Negative Samples (in total)	769	791	750

Table 4.8: Analysis of the agreements among the expert annotators and the automatically generated corpus

	Between Annotator Groups 1 and 2	Between Annotator Group 1 and the Automatically Generated Corpus	Between Annotator Group 2 and the Automatically Generated Corpus
Agreed Positive Samples	706	715	701
Agreed Negative Samples	765	750	750
Cohen's κ	0.96	0.95	0.93

F-1 score, Matthews Correlation Coefficient (MCC) and Balanced accuracy (BAcc) are also used to assess the performance of the models.

4.2.5.1 Experimental Setup

Sent2Vec was trained with various parameter settings. The cutoff value and the best model are chosen based on the MCC and BAcc over the validation set. The best hyper-parameter settings for Sent2Vec are: 500d sentence embedding, window size 20, learning rate 0.2, negative sampling loss function and sampling threshold 0.0001. For the four sequential models: hierarchical CNN (hCNN), Bi-LSTM with max pooling, hierarchical and inner attentions over Bi-LSTM; the learning rates (LR) were initialized to 0.1. With a drop in validation accuracy, the LR is multiplied by 0.2. The batch size and LR threshold are set to 50 and 0.0001, respectively. For training, stochastic gradient descent is used as the optimizer. For hCNN, 4 layers of convolution are used followed by max-pooling. Four context vectors are used for both hierarchical and inner attention mechanisms to focus on 4 distinct parts which are concatenated for final sentence representations. For all of the tree-structured transformer models, 6 parallel heads are used with 50d query, value and key matrices where 6 position-wise convolution layers are used for multi-branch attention. Two layers of CNN (first layer: 341 1d kernel and no dropout, second layer: 300 1d kernels, 0.1 dropout) are used in the PCNN layer as the composition function which is the same as Ahmed et al. [6]. For parameter tuning, Adagrad [59] with LR 0.0002 is used in all cases.

4.2.5.2 Performance Analysis

We first evaluate the quality of the automatically generated corpus. For analyzing the quality of the data annotation, we randomly picked 750 positive and 750 negative samples (labelled as such in the automatically generated corpus) from the 74,568 citing and candidate cited sentence pairs. These 1500 sentence pairs were provided to two groups of expert annotators. Each group consisted of three people and each person annotated 500 samples. So, each 500 sample chunk was annotated by two individuals, one from each group. Each reviewer also mentioned their confidence level for each sample annotation. We then used Cohen's κ [52] to compute inter-annotator reliability between the human annotators and the automatically generated corpus. The overall statistics are shown in Table 4.7. The first group identified 731 positive and 769 negative samples in the 1500 sentence pairs, and the second group identified 709 positive and 791 negative samples. Table 4.8 shows the annotator groups' decisions agreed for 706 positive and 765 negative samples. The reliability factor κ found here is 0.96. While comparing the annotation provided by the automatically generated corpus against the first and second annotator groups, we see that the annotation decisions match for 715 and 701 positive samples between the automatically generated corpus and groups 1 and 2, respectively. For negative samples, the agreed decisions are 750 samples in both cases. The κ values are 0.95 (between first annotator group and the automatically generated corpus) and 0.93 (between second annotator group and the automatically generated corpus). These values indicate that the automatically generated corpus annotations match the experts' annotations quite well. When interpreting these high κ values, it is important to recall that the data given to the annotators were balanced (50/50 split of positive and negative samples). From Table 4.8 it is clear that the human annotators have high agreement for both of their positive and negative choices.

Next we provide the citation linkage task outcomes. To compare the performance of the model against the previous models, we evaluated the model with the gold standard human annotated data from Houngbo and Mercer [85] because the previous models were tested against this gold standard corpus. This corpus focuses on citations of methods used in the citing and cited articles. Houngbo and Mercer [86] suggests that in most cases the citation refers to single sentences in the cited articles. As an example, the citing statement "Recently, Chauhan et al. employed SVM to predict the ATP binding residues in ATP binding proteins using amino acid sequences and their evolutionary profiles" [65] indicates the cited sentence "Our SVM module predicts a score for each residue in protein (in range of -1.0 to 1.0), we define a threshold to discriminate ATP interacting and non-interacting residues" [43]. Another approach for such a task could have been ranking the candidate sentences as was one of the methods done by Houngbo [86]. However, for the final classification step we used `softmax`, which gives a

Table 4.9: Performance analysis of different architectures for the citation linkage task for biomedical research articles. Models tagged with † are the investigated ones in this work. Here, CT: constituency tree, DT: dependency tree, MB: multi-branch attention, MH: multi-head attention, TP: true-positive, FP: false-positive, TN: true-negative, FN: false-negative.

	Model	TP	FP	TN	FN	F1	MCC	BAcc (in %)
Previous Works	Houngbo	34	995	2001	27	0.06	0.07	61.27
	Li	39	779	2217	22	0.09	0.12	68.97
Sequential Models	Hierarchical CNN †	45	580	2416	16	0.13	0.19	77.21
	Bi-LSTM + Max-pooling †	54	361	2635	7	0.23	0.31	88.24
	Inner attentive Bi-LSTM †	55	372	2624	6	0.23	0.31	88.87
	Hierarchical Attentive Bi-LSTM †	56	355	2641	5	0.24	0.33	89.98
Tree Structured	DT-Transformer (MH) †	57	301	2695	4	0.27	0.36	91.70
	DT-Transformer (MB) †	58	287	2709	3	0.29	0.38	92.75
	CT-Transformer (MH) †	57	315	2681	4	0.26	0.35	91.46
	CT-Transformer (MB) †	57	309	2687	4	0.27	0.36	91.56
Ensemble	Winner-takes-all ensemble †	59	253	2743	2	0.32	0.41	94.14
	BioCite †	**60**	**240**	**2756**	**1**	**0.33**	**0.42**	**95.17**

probability to every possible outcome, so this approach could easily be modified to be a ranking approach.

Table 4.9 reflects multiple performance metrics found for the models used here along with the results from a few prominent works. Among the sequential models, Bi-LSTM with the hierarchical attention mechanism fed with Bio-RoBERTa embeddings performs the best based on the MCC and BAcc (0.33 and 89.98% accordingly). However, it can correctly extract only 56 out of 61 positive samples. The inner attentive Bi-LSTM and simple Bi-LSTM followed by a max-pooling layer captures 54 and 55 positive samples correctly with the same MCC (0.31) and F1 score (0.23). However, the inner attentive Bi-LSTM model earns a slightly higher BAcc (88.87%) as it predicts more negative samples correctly.

The tree-structured models outperform all of the sequential models to extract the cited statements from the referenced documents. The reason for this is the constituency tree-transformer is able to capture phrase level information and the dependency tree-transformer is able to preserve word level dependencies. In biomedical articles, biological components' chemical names may comprise multiple words. The constituency tree-transformer has the capability to work better with such phrase level text. And in a lot of cases, the citing statements are complex in nature. The dependency tree-transformer deals with such cases well. Another important thing to notice here is that tree-transformers with multi-branch attention perform better than the tree-transformers with multi-head attention as multi-branch attention applies multiple heads in each branch and is thus able to obtain more information about each sentence [62]. Here, both the

constituency and dependency tree-transformers with multi-head attention mechanism predict 57 positive samples correctly. Multi-branch attentive dependency tree-transformer predicts 58 positive samples correctly. Constituency tree-transformer with multi-branch attention predicts 57 positive samples correctly. However, it predicts 6 more negative samples correctly attaining a 0.10 percentage point improvement in BAcc.

The two ensemble architectures investigated here improve the performance of the citation linkage task for biomedical research articles. The first approach ensembles all of the investigated individual models with the winner takes all selection process. This approach considers all the outcomes from different models and the outcome with the highest probability is chosen as the final prediction. It successfully predicts 59 positive samples out of 61 with 94.14% BAcc, 0.41 MCC and 0.32 F1 score which are higher compared to any of the standalone models. The second ensemble architecture considers only dependency and constituency tree-transformers with multi-branch attention. There are two reasons behind choosing only these two models for ensemble in this case: firstly, the major intention was to investigate how the model performs if we combine both the word dependency and phrase level information, and secondly, these two models showed better performance among all individual models. This ensemble architecture extracts 60 true positive cited statements given the citing statements for the citation linkage task. It also achieves 95.17% BAcc, 0.42 MCC and 0.33 F1 score. As the best performance is attained by this last ensemble architecture, for the BioCite citation linkage framework, we choose this approach for extracting cited statements from the referenced biomedical research article given the citing statement from the citing paper. Is the computationally more expensive ensemble model justified for predicting only a few more true-positives? We notice that the increase in true-positives is approximately 2%. This increase, especially in a larger corpus, would seem to justify the extra computational cost. However, it should also be noted that the false-positives have decreased by almost 20%. The applications noted in the introduction will benefit substantially by such a decrease in false-positives. This decrease in false-positives further justifies the extra computational cost of the ensemble model.

Now, there remains one more question to be discussed. Which one is actually improving the performance, the automatically generated corpus or the model? From Table 4.9, it is clear that, the performance of BioCite is better than the other models. To check the effectiveness of the proposed automatically generated corpus, we trained all the models over the human annotated small corpus [85]. In this experiment we found all the investigated models' accuracies were very high (around 98%). However, the BAcc, MCC and the F1 scores were very poor as the models are strongly biased towards the negative outcome. This gives evidence of the effectiveness of training models over our proposed automatically generated corpus. Furthermore, analyzing the outcomes and going through the predictions of the sentence pairs, we

found that this model can successfully predict cited sentence given the citing statement when chemical components and reactions are presented in different ways. For example: the cited sentence “DNA is fairly stable in mildly acidic solutions, although the beta glycosidic link in the purine bases is hydrolyzed at around pH4.” [29] is predicted successfully for the citing sentence “DNA samples are frequently harmed by exposure to excessively acidic environment.”, [230]. It indicates that this model has the ability to resolve “pH4” as an “excessively acidic environment” and “hydrolyzed” with “harmed”.

4.2.6 Conclusion

Biomedical literature is complex in nature due to having complex biological and chemical component names. Our framework, BioCite, performs well when dealing with the human annotated test set containing research articles accumulated from the biomedical domain and outperforms the previous prominent works. However, there are still a few avenues to investigate. The text span used here is a single sentence. In future, it can be expanded to the paragraph level which would capture the contextual information as well. Graph-based neural networks which perform well when working with paragraphs [268] could be used. Moreover, BERT-based models can be explored as well.

4.3 Conclusion

The current method of citing in scientific articles is to refer to the source document, but this does not reflect the text chunk from the source document that is being referred to by the citing statement. To ease the task of understanding the background information, the citation linkage framework tries to retrieve the text spans from the referenced articles that are the focus of the citations in the citing article. This task has been formulated as a semantic similarity measurement task since a citing statement can be an interpretation of the cited text span, or a paraphrase of it, or an explanation of the text written in the reference article. However, we had to restrict this task to the sentence level due to the lack of a test corpus that is human-annotated beyond the single sentence level. In addition, our proposed approach for creating the silver standard corpus demands a gold standard corpus for validation. For the sentence level citation linkage task, experiments with different models show the effectiveness of training with our proposed silver standard corpus. And the ensemble architectures proposed here have also shown superior performance compared to the other approaches.

For the negative sentence selection we applied the algorithm we have mentioned in section 4.1. Following this algorithm we randomly selected negative samples three times and every

time the performance of the models were almost identical. Finally, the best negative samples were selected via experiments over the validation set from the gold standard corpus. Rather than analyzing how close the negative samples are to the positive samples in the vector space, we simply assumed that they were different enough depending on the performance over the validation set.

No separate ablation study of the ensemble models has been presented here as the performances of the individual models, which have been combined together to build the ensemble architectures, have been reported. Reporting the results of the individual models serves as an alternative to an ablation study.

As these models were implemented in early 2020, experimenting with the large language models (LLMs) was not possible due to resource limitations. Initially, we performed a few experiments with the pretrained sentence-BERT (SBERT), but the performance was not satisfactory as SBERT was not trained on biomedical texts. With the limited resources, it was not possible to fine-tune the SBERT model. Experimenting with LLMs can be explored in the future to tackle this task. As well, the heterogeneous graph attention network that has been used in the following chapters for word embedding enrichment can be applied here to see if this idea can improve performance.

Another dimension of future work can be to expand the retrieved text span to beyond the sentence level. The initial task for that approach would be to create a gold standard corpus. If that corpus contains a small number of samples, then our proposed approach for synthetic corpus creation could be used to make a proper sized silver standard corpus. Moreover, the current gold standard corpus deals with method citations only. Future work could extend the set of citation types to the kinds of citations that refer to results, motivations, and so forth.

Chapter 5

Biomedical Entity Relation Extraction

The objective of relation extraction is to discern the interaction between two entities within a text. Our focus in this chapter lies in exploring this extraction task to detect interactions between various proteins and drugs mentioned in a sentence. This chapter amalgamates the findings from three of our publications: (i) **"Investigating Protein-Protein Interactions using Tree-Structured Neural Network Models"**, (ii) **"Identifying Protein-Protein Interaction using Tree-Transformers and Heterogeneous Graph Neural Network"**, and (iii) **"Extracting Drug-Drug and Protein-Protein Interactions from Text Using a Continuous Update of Tree-Transformers"**.

Our evolving approaches are motivated by the idea of incorporating syntactic information to preserve phrasal and inter-word relational details for this task. Initially, we utilized only tree transformers and an ensemble thereof for sentence vector representation. Subsequently, we integrated the graph attention network to generate an additional sentence representation, which is then updated with context-enriched word representations. Our most recent work further updates the word embeddings of the tree transformers with context-enriched word embeddings, yielding the best performance. This approach has demonstrated state-of-the-art performance in comparison to previous methods and has maintained its position as the benchmark for tasks involving protein-protein and drug-drug interactions.

The standard corpora that are used in these studies are modified by replacing the protein and drug names with some out-of-vocabulary generic terms. All of the models are fed with BioRoBERTa word embeddings, as BioRoBERTa is trained on biomedical texts and also has the ability to generate vector representations for out-of-vocabulary words. Our initial experiments used fasttext and BioWordVec word embeddings, both being able to provide vector representations for out-of-vocabulary words. However, the better results were found with BioRoBERTa and that is why all the experiments reported here are done with BioRoBERTa word embeddings.

5.1 Investigating Protein-Protein Interactions using Tree-Structured Neural Network Models

This section is based on the paper titled "Investigating Protein-Protein Interactions using Tree-Structured Neural Network Models" co-authored with Robert E. Mercer that appeared in *The 35th International FLAIRS Conference Proceedings (FLAIRS 35)* [204]. This paper was nominated for best student paper award and secured the runner-up position.

In order to comprehend underlying biological processes, it is necessary to identify interactions between proteins. It is typically quite difficult to extract a protein-protein interaction (PPI) from text data as text data is complex in nature. Unlike sequential models, tree-structured neural network models have the ability to consider syntactic and semantic dependencies between different portions of the text and can provide structural information at the phrase level. This paper investigates tree-structured neural network models for the PPI task and the results show their supremacy over sequential models and their effectiveness for this task.

5.1.1 Introduction

As the scientific literature grows at an exponential rate, the vast majority of biological information is currently available in text form residing in the scientific literature. MEDLINE database's size has grown by 4.2 percent annually during the last two decades and currently it contains approximately 26,000,000 records extracted from 5639 publications which is 23% more than what it contained in 2014 [245]. This huge amount of unstructured text from biomedical research articles is a valuable source of information for the biomedical natural language processing (NLP) domain.

As the volume of biomedical data continues to grow exponentially and due to the inherent complexity in the textual representations of these data, it is critical to pursue automatic information retrieval techniques to aid biologists in the detection and identification of useful information, and the arranging and maintaining of databases, as well as providing automatically generated decision support systems for medical professionals. Considering this issue, a lot of research has been conducted for inferring information concealed in these texts to assist health care and biomedical people, such as protein-protein interactions (PPIs), chemical-disease relation extraction, clinical relation extraction, drug-drug interactions, etc., as retrieving important information manually from this large volume of texts is both time consuming and expensive [165].

The majority of biological activities inside a cell, such as immune response, signal transduction, cellular organization, etc., are caused by different interactions between various pro-

teins [209]. So, identifying the protein-protein interactions (PPIs) provides a better understanding of the functionalities, regulations, and communication between different proteins [250]. Identifying PPIs entails figuring out how different proteins mentioned in a text are connected [112]. This information may spread out through different parts of the whole document, however, the current work is restricted to identify PPIs present only inside single sentences [172, 218]. For instance, "LEC induced maximal migration of CCR1 and CCR8 transfected cells at 89.3 nmol/L and cell adhesion at 5.6 nmol/L." [87] reflects two PPI relations: LEC-CCR1 and LEC-CCR8 and no association between CCR1 and CCR8.

For such tasks, sequential deep learning-based models have been used in multiple research works [88, 244]. However, if the data is structured rather than presented sequentially, these models are more likely to miss the underlying semantic compositionality [5] as they consider word order only but no linguistic structure [118]. By contrast, Recursive neural networks, commonly known as tree-structured neural network models, work over parsed tree representations of the sentences and thus preserve both the syntactic and semantics in a better way.

In this paper, we investigate six tree-structured neural network models for the PPI identification task. For working with dependencies between words in different portions of the sentence, we investigate dependency tree-structured neural nets, whereas to work with the phrase level information, constituency tree-structured neural nets are explored. Finally, two ensembles of these models are used for retrieving the PPIs present in the sentences. We provide an ample analysis of these models' performances over the benchmark PPI datasets which evince the supremacy of using tree-structured neural networks over sequential ones for this task.

5.1.2 Related Works

Numerous NLP methods have been developed for determining the associations between protein entities. In the initial stages, pattern-based techniques were widely used. In these approaches, on the basis of syntactic as well as lexical features, pattern-based rules were designed for extracting the relationship [27, 114]. However, these models were not capable of handling complex relationships specified in relational and coordinating clauses appropriately. In contrast to naïve pattern-based methods, dependency-based methods are more syntax attentive and offer a wider range of application coverage [60, 150].

Another prominent approach for extracting such relationships is to use kernel-based methods. These models learn profuse structural information by means of dependency structures and syntactic parse trees [149, 109]. Some noteworthy methods use bag-of-words kernel [191], tree kernel [263], convolution tree kernel [42], neighbourhood hash graph kernel [266], and walk-weighted kernel [109].

With the recent blossoming of deep learning-based models, a lot of experiments have been conducted to extract PPI relations [175, 270]. Peng et al. [164] deployed a double-channel convolutional neural network (CNN) for feature extraction. The first channel utilizes syntactic features like named entities, parts of speech, syntactic dependencies, chunk parsing information, distances from each word to the two interacting protein candidates, and the word itself. The second channel applies a convolution operation over each word's parent word information. Zhang et al. [267] applied a three channel CNN for this task. The first, second, and third channel apply convolutional operations over original words in addition to the positional encoding, shortest dependency path information, and dependency relation encoding features, respectively. Zhao et al. [270] trained an auto-encoder on the unlabelled training data for parameter initialization of a multi-layer perceptron (MLP) model which is then trained utilizing gradient descent for the PPI relation extraction task.

Following this, several research works have been conducted for this task using recurrent neural networks (RNNs) as these models perform better with sequential data. Hsieh et al. [88] applied only a bi-directional long short term memory network (Bi-LSTM) on the sentences, and the vectors concatenated from the left and right-most LSTM output vectors are used as the feature vectors for the classification task. Yadav et al. [244] utilized structured attention over the sequential Bi-LSTM which is fed with the shortest dependency path information between the unit pairs. In their following work, Yadav et al. [245] utilized the self attention mechanism for multi-task learning incorporating both PPI and drug-drug interaction relation extraction. Ahmed et al. [4] used a dependency tree-structured LSTM with structured attention for the same task and outperformed all of the above-stated sequential models.

5.1.3 The Model

This section describes our work in detail. Our investigation of the PPI relation extraction task examines four tree-structured neural network models: dependency and constituency tree-LSTMs with a self attention mechanism and tree-transformers. Their working principles are discussed first. Two ensembles of these models are then discussed.

5.1.3.1 Tree-LSTMs

A sentence can be represented by two tree-structured representations: constituency and dependency trees [44]. These representations provide syntactical information about the sentence by preserving word to word dependencies (dependency tree) and phrase level information (constituency tree). To utilize these structural syntactic information sources, Tai et al. [214] introduced dependency (child sum tree-LSTM) and constituency (N-ary tree-LSTM) tree-LSTMs.

For the child sum tree, the internal gates of a component node are updated by the summed hidden state values of its child nodes. Then, using this updated hidden state value the other intermediate gates are updated as follows:

$$\tilde{h_j} = \sum_{\kappa \in C_j} h_{j\kappa} \tag{5.1}$$

$$i_j = \sigma(W_i x_j + U_i \tilde{h_j} + b_i) \tag{5.2}$$

$$o_j = \sigma(W_o x_j + U_o \tilde{h_j} + b_o) \tag{5.3}$$

$$\tilde{c_j} = \tanh(W_c x_j + U_c \tilde{h_j} + b_c) \tag{5.4}$$

Here, Ws and bs are weights and bias values, and C_j is the set of child nodes. In the child sum tree-LSTM, for each child node, there is a separate forget gate ($f_{j\kappa}$) which allows the model to selectively incorporate information for the parent node from the child nodes. For each child node, the corresponding cell state and forget gate values are then multiplied and finally all of these values are combined together to compute the forget gate value of the parent node. Then, the cell state (c_j) and hidden state (h_j) values of the parent node are computed using this forget gate value as follows:

$$f_{j\kappa} = \sigma(W_f x_j + U_f \tilde{h_j} + b_f) \tag{5.5}$$

$$\tilde{f_j} = \sum_{\kappa \in C_j} f_{j\kappa} \cdot c_\kappa \tag{5.6}$$

$$c_j = i_j \cdot \tilde{c_j} + \tilde{f_j} \tag{5.7}$$

$$h_j = o_j \cdot \tanh(c_j) \tag{5.8}$$

In the N-ary tree-LSTM, each parent node contains identical cell and hidden states for each of its children. The internal gate values and forget gates are computed as follows:

$$i_j = \sigma(W_i x_j + \sum_{\iota=1}^{N} U_{i\_\iota} \tilde{h_j} l + b_i) \tag{5.9}$$

$$o_j = \sigma(W_o x_j + \sum_{\iota=1}^{N} U_{o\_\iota} \tilde{h_j} l + b_o) \tag{5.10}$$

$$\tilde{c_j} = \tanh(W_c x_j + \sum_{\iota=1}^{N} U_{c\_\iota} \tilde{h_j} l + b_c) \tag{5.11}$$

$$f_{jk} = \sigma(W_f x_j + \sum_{\iota=1}^{N} U_{f\_j\iota} h_j l + b_f) \tag{5.12}$$

Just like in the child sum tree-LSTM, the final forget gate of the parent node is computed by multiplying the corresponding forget gate and cell state values and then summing them (Eq. 5.13). The cell state (Eq. 5.7) and new hidden state (Eq. 5.8) values are computed as before.

$$\tilde{f}_j = \sum_{\iota=1}^{N} f_{j\iota} \cdot c_{j\iota} \tag{5.13}$$

Ahmed et al. [5] introduced self attention for such tree structured recursive neural networks. It incorporates three matrices: *query, key,* and *value*. They are calculated as follows:

$$key = \omega_k \mathcal{M}_k \qquad \text{s.t.} \quad \omega_k \in \mathbb{R}^{d \times d} \tag{5.14}$$

$$value = \omega_v \mathcal{M}_v \qquad \text{s.t.} \quad \omega_v \in \mathbb{R}^{d \times d} \tag{5.15}$$

$$query = \omega_q \mathcal{M}_q \qquad \text{s.t.} \quad \omega_q \in \mathbb{R}^{d \times d} \tag{5.16}$$

For the child sum tree, the $\mathcal{M}$s are the concatenations of all of the child nodes' word vectors for a corresponding parent node, whereas in the N-ary tree-LSTM the word vectors under a constituent are concatenated. Then these *key, value,* and *query* matrices are aligned considering the representation's dimension (Eq. 5.17).

$$align \in \mathbb{R}^{n \times n} = (query)^T key \cdot (1/\sqrt{d}) \tag{5.17}$$

where n is the number of offspring nodes under any particular parent node and d is the normalizing factor. Then, `softmax` is applied over this *align* matrix to compute the attention probability matrix $\alpha \in \mathbb{R}^{n \times n}$. Finally, batch-wise matrix multiplication is applied between the attention matrix α and the matrix *value* to compute the attentive hidden states $\tilde{h} \in \mathbb{R}^{n \times d}$. Rows of this matrix are concatenated to produce the final hidden representation of the parent node for both the child sum and N-ary tree-LSTM.

5.1.3.2 Tree-Transformers

Ahmed et al. [5] applied the concept of transformer [223] over the constituency and dependency trees, and introduced two tree-transformer models: constituency tree-transformer and dependency tree-transformer. Both of these models apply multi-branch attention over the child nodes' representations. Just like the self attention mechanism, this approach also uses *key,*

query, and *value* matrices as follows:

$$\alpha = \texttt{softmax}(\frac{query\ key^T}{\sqrt{d_k}})value \tag{5.18}$$

where d_k is the dimension of the *key*. For the multi-branch attention (β_i), n copies of *key, query,* and *value* matrices are created with the appropriate weight matrices ω_i, where n is the number of branches, and finally a scaled dot product attention (Eq. 5.18) is applied over each branch (Eq. 5.19).

$$\beta_i = \alpha_{i\in[1,n]}(query_i\ \omega_i^{query}, key_i\ \omega_i^{key}, value_i\ \omega_i^{value}) \tag{5.19}$$

A residual connection is then employed over these tensors followed by a layer-wise batch normalization layer. A scaling factor τ is applied in the end to produce the branch representation (Eq. 5.20). Following this, position-wise CNN (PCNN) is applied over every $\tilde{\beta}_i$ (Eq. 5.21). The attention-encoded representations of these semantic subspaces are computed by applying weighted summation where each $\gamma_i \in \mathcal{R}^n$ is a hyperparameter (Eq. 5.22). In the end, with `BranchAttn`, another residual connection is employed. This is then fed to a tanh layer and an element-wise summation (`EWS`) is performed to generate the parent node representation (Eq. 5.23). Here, χ and $\tilde{\chi}$ represent the input and the outcome of the attention module, respectively.

$$\tilde{\beta}_i = \texttt{LayerNorm}(\beta_i\omega_i^b + \beta_i) \times \tau_i \tag{5.20}$$

$$\texttt{PCNN}(x) = \texttt{Conv}(Relu(\texttt{Conv}(x) + b_1)) + b_2 \tag{5.21}$$

$$\texttt{BranchAttn} = \sum_{i=1}^{n} \gamma_i \texttt{PCNN}(\tilde{\beta}_i) \tag{5.22}$$

$$\texttt{ParentNodeRep} = \texttt{EWS}(\tanh((\tilde{\chi} + \chi)\omega + b)) \tag{5.23}$$

5.1.3.3 Ensemble Architecture

After exploring the tree-structured LSTMs and transformer models, we investigated two ensemble models. In the first approach, we train all the models, and then, when testing, each sentence is fed to all of the models. All of the models predict the class label individually. Finally, a winner takes all method [188] is applied over these individual models' selections for the final class prediction. In our second approach, we utilize only the dependency and constituency tree-transformers. Each sentence is fed to both of the tree-transformers and then the sentence representations are concatenated and then fed to the following MLP for class label prediction. The intention behind investigating this model is to find out what happens if features containing both word-level dependencies and phrase-level information are used for the

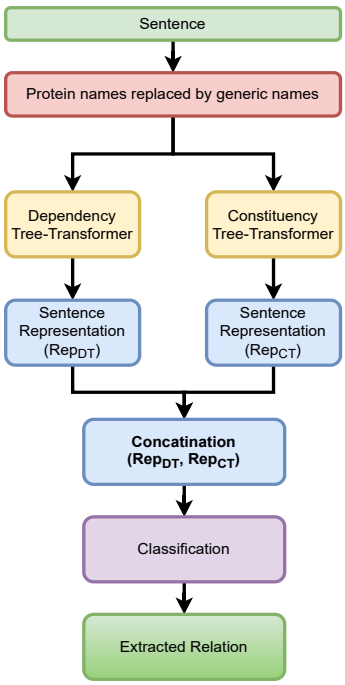

Figure 5.1: Working procedure of the ensemble architecture combining features from the dependency and constituency tree-transformers.

PPI relation extraction task. Figure 5.1 provides a sketch of this ensemble architecture.

5.1.4 Experiments and Performance Analysis

This section reports the results found for the tree-structured neural network models and the ensemble architectures with F1-score as the performance evaluation metric. It also provides a statistical description of the five standard benchmark PPI corpora for this task along with the pre-processing steps. The PPI problem has been formulated as a classification task. Finally, the performance of the tree-structured models are compared against the most prominent sequential and the previous tree-structured architectures used to solve this task.

For evaluating the investigated tree-structured neural networks, the models are tested on the five standard PPI corpora: AIMed [35], BioInfer [173], IEPA [57], HPRD50 [69], and LLL [158]. For all the experiments, the converted version of the corpora are used as mentioned by Ahmed et al. [5]. All of the protein names in all five corpora are substituted with three special symbols: PROT0, PROT1 and PROT2. If any two proteins in a sentence are being considered as interacting with each other, they are replaced with PROT1 and PROT2. All the mentioned proteins in the sentence which are not being considered for interaction identification are replaced with PROT0. As an example, the sentence “LEC induced maximal migration of CCR1 and CCR8 transfected cells at 89.3 nmol/L and cell adhesion at 5.6 nmol/L.”, the protein names

Table 5.1: Overall demographics of the modified corpora

Corpus	Original Sentences	Positive Interactions	Negative Interactions
AIMED	1,995	1,000	4,834
BioInfer	1,100	2,534	7,132
IEPA	486	335	482
HPRD50	145	163	270
LLL	77	164	166

LEC, CCR1 and CCR8 are replaced by PROT1, PROT2 and PROT0 accordingly as this time the intention is to retrieve the relation between LEC and CCR1. When the target proteins are LEC and CCR8, then these two protein names are replaced by PROT1 and PROT2 accordingly, and CCR1 would be replaced with PROT0. The nature of an interaction between two proteins can be positive or negative. For the above mentioned two examples, the interactions are positive whereas the interaction between CCR1 and CCR8 is negative as there is no interaction between them. There are three possible interactions present in this example sentence. So, the modified corpora contains three variants of this sentence with two positive and one negative interaction. In a similar way, for every sentence in the corpora with η proteins present in it, there are ${}^{\eta}C_2$ variants in the modified corpora. The demographics of these five modified corpora are presented in Table 5.1. In addition, representing the protein names by a few generic names enhances the data further by having multiple samples for these generic names rather than a few samples for each real protein name. For the evaluation of these models, we used 10-fold cross validation using StratifiedK-Fold from the scikit-learn package.

Both of the tree-LSTM models are initialized with learning rate 0.1. For each iteration, if the validation accuracy drops compared to the previous iteration, the learning rate is reduced by 80%. The batch size is 10. The memory and attention dimension is set to 150. The MLP hidden dimension is 300. Training dropout is used with value 0.1. For the training of the tree-LSTM models the 'SGD' optimizer is used. For the tree-transformer models, the initial learning rate is 0.1 and the same learning rate decay approach is used. Six PCNN layers are used in the multi-branch attention block. For the experiments, six branches of attention layer are used. Each PCNN layer is composed of 2 CNNs. The first CNN layer employs 341-dimension kernels without any dropout and 300-dimension kernels are used in the second layer of the CNN. In the second layer, dropout 0.1 is used in all cases just like Ahmed et al. [5]. The hyperparameters of the tree-transformers are updated using the 'Adagrad' optimizer. All of the models (both tree-LSTMs and tree-transformers) are fed with Bio-RoBERTa [82] word embeddings which are not updated during training. We also tried fasttext [28] and Bio-WordVec [265] embeddings. However, the best results are with the Bio-RoBERTa embeddings.

Table 5.2: Performance evaluation of the models by means of F1-score (in %). The sequential models are marked with †. Here, NT-LSTM: 2-ary tree-LSTM over constituency tree, CT-LSTM: Child sum tree-LSTM over dependency tree, CT-Transformer: Constituency tree-transformer and DT-Transformer: Dependency tree-transformer, DT+CT-Transformer: Combination of dependency and constituency tree-transformers.

Methods	AIMed	BioInfer	IEPA	HPRD50	LLL	Avg.
Chang et al. [42] †	60.6	69.4	71.4	71.5	80.6	70.7
Hsieh et al. [88] †	76.9	87.2	76.31	80.51	78.3	79.84
Zhang et al. [267] †	56.4	61.3	75.1	63.4	76.5	66.54
Yadav et al. [245] †	77.33	76.33	-	-	-	76.83
Tai et al. [214]	80.6	88.1	76.4	82.0	84.8	82.38
Ahmed et al. [4]	81.6	89.1	78.5	81.3	84.2	82.94
NT-LSTM + Self Attn	82.99	90.87	78.2	83.22	86.14	84.28
CT-LSTM + Self Attn	83.06	91.01	78.9	83.59	86.78	84.67
CT-Transformer	87.51	94.95	82.5	87.73	91.32	88.80
DT-Transformer	87.88	95.37	82.56	88.01	91.46	89.06
Ensemble - Winner Takes All	87.94	95.48	82.63	87.95	91.49	89.09
DT + CT-Transformer	**88.15**	**96.01**	**83.24**	**88.94**	**92.18**	**89.70**

Table 5.2 shows the performance of the tree-structured LSTMs, transformers, and the ensemble architectures over the five benchmark PPI corpora and some prominent sequential and tree-structured models for comparability. Among these five, AIMed contains many erroneous annotations. In addition, having nested named entities makes it more difficult to work with [4].

From this table it is clearly visible that all of the tree-structured models outperform the sequential models for this task. Among the two investigated tree-LSTM models (child sum and N-ary treeLSTM), the child sum tree-LSTM with self-attention performs slightly better than the N-ary tree-LSTM with self attention (average F1-scores are 84.67% and 84.28% accordingly). Overall, both of the investigated tree-transformer models perform better than the tree-LSTM models. However, among the tree-transformer architectures, the dependency tree-transformer (DT-transformer) performs better than the constituency tree-transformer (CT-transformer) and it happens for all five corpora. So, these two observations suggest that neural networks based on dependency trees perform slightly better than the models built on constituency trees. The reason behind this may be that because the sentences here are quite complex in nature, word-level dependency provides more useful information.

For each dataset among these four standalone models, DT-Transformer has the highest F1-scores (87.88%, 95.37%, 82.56%, 88.01%, and 91.46% for the AIMed, BioInfer, IEPA, HPRD50, LLL datasets, accordingly). For the Ensemble - Winner Takes All model, a little performance boost is achieved for four datasets. For HPRD50, the F1-score is a bit less than the two transformer models. It achieves a better average F1-score compared to all of the standalone

models. The DT+CT-Transformer model combines features from both of the constituency and dependency tree transformers. The reasons behind choosing only the tree-transformer-based models are that both of the transformer-based models perform better than the LSTM-based models and both the word dependency-level and phrase-level information are already being provided by the transformer-based models. The DT+CT-Transformer outperforms all other models for every dataset with an average F1-score 89.70%. Furthermore, this approach is computationally less expensive compared to the previously mentioned ensemble model as that method requires four models to be trained whereas for the DT+CT-Transformer only two models and an additional MLP are required to be trained. Additionally, the results can be explained by means of the attention value on each node as presented by Ahmed et al. [5] .

5.1.5 Conclusions

In this work, we have explored various tree-structured neural network models for the PPI relation extraction task. The experimental results show that the tree-structured models, because of having additional syntactical information at word dependency and phrase-level, perform better than the sequential models. Among all of the explored models, the combined model with both the dependency and constituency tree-transformers performs the best as it utilizes both the word dependency and constituency information. However, opportunities for improvement in this field remain. In the future we want to explore graph-based neural network models with attention mechanisms, and to leverage additional features for this task. Further analysis of results based on AUC and ROC curves can be performed.

5.2 Identifying Protein-Protein Interaction using Tree-Transformers and Heterogeneous Graph Neural Network

This section is based on the paper titled "Investigating Protein-Protein Interactions using Tree-Structured Neural Network Models" co-authored with Robert E. Mercer that appeared in *The 36th International FLAIRS Conference Proceedings (FLAIRS 36)* [201].

For a better understanding of the underlying biological mechanisms, it is crucial to identify the reciprocity between proteins. Often, extracting such interactions between proteins from biomedical articles faces challenges due to the complex sentence structure of the textual information sources. Most of the prominent previous works have applied additional hand-crafted features for the protein-protein interaction task. In this work, we have utilized two tree-

structured attention-based neural network models along with a heterogeneous graph approach to perform this task. We suggest that the proposed model preserves the syntactic as well as the semantic information of the text. The experimental results demonstrate that even without using any additional feature extraction techniques, this model achieves significant performance boosts when applied on the five standard benchmark corpora compared to the previous works.

5.2.1 Introduction

The exponential growth of scientific literature means that the majority of biological information is now in text form and can be found in the scientific literature. The MEDLINE database has seen an increase of more than 4% each year over the past two decennia, and currently holds more than 29 million records from various publications, which is 3 million more than in 2020 and more than 8 million over what it held in 2014 [245]. The massive amount of text found in biomedical research articles represents an invaluable resource of information for the field of automated biomedical information retrieval.

Given the exponential growth of biomedical data and the intricate nature of the textual representation of this data, it is crucial to utilize automated methods for information retrieval to assist biologists in finding relevant information, organizing databases, and offering decision support for medical professionals. Several studies have been conducted to extract the information present in these texts, including protein-protein interactions, chemical-disease relationships, clinical relations, drug-drug interactions, and more.

A cell's internal biological activities, including immune response, signal transduction, and cellular organization, are largely a result of interactions between various proteins [209]. Understanding molecular mechanisms of biological processes requires knowledge of protein-protein interactions (PPI) [4]. These interactions have crucial relevance for biomedical fields, including the examination of drug targets [77] and signal proteins [9]. Therefore, recognizing protein-protein interactions (PPIs) leads to a deeper comprehension of the functions, control, and communication between various proteins [250]. The objective of recognizing PPIs is to extract the relationships between protein entities mentioned in a document [112].

A significant amount of information regarding PPIs is present in biomedical literature, but in an unstructured form. Manually extracting PPIs is a demanding task, both in terms of time and cost, due to the large number of published studies [165, 216]. As a result, automatically extracting PPIs from biomedical literature has become a crucial research area, garnering attention from many researchers. The information could be dispersed throughout the document, however, the current study is limited to detecting only the PPIs within individual sentences similar to many previous works [172, 218, 4]. As an example of a sentence containing interactions

between proteins [87]:

> "At 89.3 nmol/L, maximal migration of CCR1 and CCR8 transfected cells was prompted by LEC and at 5.6 nmol/L, cell adhesion also occurred."

This sentence reflects two protein-protein interactions involving LEC and CCR1, as well as LEC and CCR8. But, importantly, no correlation is present between proteins CCR1 and CCR8.

In the early research phase, the commonly used methods for PPI extraction involved utilizing co-occurrence and pattern recognition techniques [20, 256]. However, recent advancements in technology have led to the widespread adoption of machine learning techniques which have superior performance compared to these traditional methods. Early approaches involved constructing a feature set through feature engineering and kernel methods and then applying support vector machines or other classifiers for classification [7, 152]. In the last few years, several research works [270, 89, 49] have successfully applied deep learning techniques to PPI extraction, taking advantage of the widespread use of deep learning in NLP.

Most of the recent works utilize recurrent neural network (RNN) models for this task considering textual representations as sequences [88, 244]. However, if the data is arranged in a structured format instead of being arranged in a sequence, these models are prone to miss the semantic compositions present within [5]. This is due to the fact that they only take into account the word order and ignore the linguistic structure [118]. Contrarily, recursive neural networks, also known as tree-structured neural network models [6], process the sentences represented in a parsed tree form, thereby keeping both the syntax and semantics in a more effective way. Investigations have also taken place regarding graph-based methods for this task, where the models operate on either a fully connected graph composed of word nodes or on text segments of phrases [64]. Our proposed model assembles these last two methods in a novel design.

While extracting relations between target proteins, we have considered three issues: firstly, how to retrieve the relation if the considered proteins are mentioned far apart in the text, secondly, how to deal with the phrasal structure of text in order to preserve the semantics so that the PPI extraction can attend to this information, and thirdly, what will happen if instead of using fixed word representations from pre-trained models, we update the word representations based on the considered sentence and then use these updated representations to impact the generated sentence representation for this task.

Uniting these considerations, we have investigated a model combining dependency and constituency tree transformers [6] and a heterogeneous graph attention network [227] for the PPI extraction task. The dependency tree-transformer captures the correlations between words at different parts of the sentence which allows the model to extract relations between the considered proteins even if they are positioned far apart in the sentence. For preserving the phrasal

information we have used the constituency tree-transformer. And for word to sentence representation and sentence to word representation updates, we have utilized a heterogeneous graph neural network. We provide a comprehensive analysis of the performance of these models on benchmark PPI datasets, which showcases the superiority of the proposed model over the previous prominent works.

5.2.2 Related Work

Several NLP techniques have emerged for determining links between proteins. At first, pattern-based methods were popular, where rules were established based on syntax and lexical features for finding relationships [27, 114]. But, these models couldn't manage complex relationships expressed in relational and coordinating clauses correctly. Unlike simple pattern-based approaches, dependency-based methods are more focused on syntax and can be applied to a broader range of situations [60, 150].

Another common method for identifying correlations between proteins is the use of kernel-based techniques. These models acquire rich structural information through dependency structures and syntactic parse trees [204]. Airola et al. [7] suggested a method for identifying interactions between target proteins by examining information from linear and dependency subgraphs. Miwa et al. [149] developed a system that incorporates a Support Vector Machine with weighted feature vectors derived from multiple corpora. Kim et al. [109] matched e-walks and v-walks on the shortest dependency path to acquire non-contiguous syntactic structures by means of a walk-weighted sub-sequence kernel for this task. Zhang et al. [266] introduced a neighbourhood hash graph kernel-based model to draw out PPIs. Chang et al. [42] used a convolution tree kernel and PPI patterns to extract interlinkages between proteins. Murugesan et al. [152] proposed the distributed smoothed tree kernel which has demonstrated substantial advancements when compared to other kernel methods for this task.

The recent surge in deep learning models has resulted in a plethora of experiments aimed at uncovering PPI relationships from biomedical literature [175, 88, 267]. Zhao et al. [270] were the first to apply deep learning in the area of PPI relation extraction. Their approach involved training an autoencoder on unclassified training data to prepare the parameters for a multi-layer perceptron (MLP) model, which was then optimized through gradient descent to carry out PPI extraction. Peng et al. [164] involved the utilization of a double-channel CNN for this task. The first channel incorporated syntax-based features like syntactic dependencies, parts of speech, named entities, the distance of each word from the two proteins interacting, chunk parsing details, and the word itself. The second channel utilized a convolution process with respect to the parent word information for each word. The second channel provides a

distributed representation of the sentence by applying convolution over each word's parent information. For PPI extraction, a three-channel CNN was implemented by Zhang et al. [267]. Convolution operations were carried out on the original words along with positional encoding, the shortest dependency path, and encoding features for dependency relations in each of the first, second, and third channels, respectively. Zhang et al. [260] showed that using residual connections improves the performance of the CNN-based models when extracting PPIs from texts.

Since then, a series of studies have been carried out on the PPI task, utilizing Recurrent Neural Networks (RNNs), which have been seen to excel in processing sequential data. Hsieh et al. [88], to generate a sentence vector representation, concatenated the left and right-most output vectors from a Bi-LSTM which was fed with the sentence, and then applied a softmax classifier for the classification task. Yadav et al. [244] fed the shortest dependency information between unit pairs as input to a Bi-LSTM with structured attention. For their subsequent study, Yadav et al. [245] implemented a self-attentive approach for performing two tasks simultaneously: extraction of protein-protein interactions and extraction of drug-drug interactions. Ahmed et al. [4] applied structured attention over dependency tree-LSTMs for this task and showed the supremacy of the tree-structured neural networks over sequential models. Fei et al. [64] introduced a span-graph neural architecture for extracting protein entity relations from biomedical texts. Their model jointly learns to identify the candidate entity spans and the correlaton between them. The entity graph is constructed by listing out probable entity span possibilities.

5.2.3 Proposed Model

In this portion of the paper, we delve into the specifics of our model. Our study of the protein-protein interaction extraction task utilizes two tree-structured neural networks: dependency and constituency tree-transformers [6]; and a heterogeneous graph attention network [227]. How each network functions is initially discussed. The discussion then moves on to cover the proposed model combining these modules.

5.2.3.1 Tree-Transformers

Two tree-based representations exist for representing a sentence: constituency trees and dependency trees. These forms of representation offer syntactic information about the sentence, capturing both the structure of phrases (constituency tree) and the dependencies between individual words (dependency tree). Ahmed et al. [6] suggested two tree-transformer models: dependency and constituency tree-transformers utilizing these sources of syntactic structure

information. The objective of these models is to traverse each sub-tree within a dependency or constituency tree structure, attentively, and derive a vector representation at its root.

Each node in a dependency tree holds a word. To traverse a sub-tree in this kind of tree, the dependency tree-transformer considers both the parent and child node representations. Conversely, in a constituency tree, only the leaf nodes hold words. The non-terminal node vectors are computed only after the sub-tree has been fully traversed. Ahmed et al. [6] applied self-attention to the sentence's dependency and constituency tree representations, incorporating *query* ($\mathcal{Q}$), *key* ($\mathcal{K}$) and *value* ($\mathcal{V}$) matrices. These matrices are computed as follows [223]:

$$\mathcal{K} = \omega_k \mathcal{M}_k \quad \text{s.t.} \quad \omega_k \in \mathbb{R}^{d \times d} \tag{5.24}$$

$$\mathcal{V} = \omega_v \mathcal{M}_v \quad \text{s.t.} \quad \omega_v \in \mathbb{R}^{d \times d} \tag{5.25}$$

$$\mathcal{Q} = \omega_q \mathcal{M}_q \quad \text{s.t.} \quad \omega_q \in \mathbb{R}^{d \times d} \tag{5.26}$$

In the dependency tree, the matrix $\mathcal{M}$ is formed by concatenating the word vectors of all child nodes for each corresponding parent node. On the other hand, for the constituency tree, $\mathcal{M}$ is the concatenation of the word vectors within a constituent. Using $\mathcal{Q}$, $\mathcal{K}$ and $\mathcal{V}$ matrices, the tree-transformer models compute the self attention matrix as follows:

$$\alpha = \text{softmax}(\frac{\mathcal{Q}\,\mathcal{K}^T}{\sqrt{d_k}})\mathcal{V} \tag{5.27}$$

where d_k refers to the dimension of $\mathcal{K}$. To implement the multi-branch attention $\mathcal{B}_i$ with n branches, n copies of *key*, *query*, and *value* matrices are generated using the appropriate weight matrices (ω_i). In the end, a scaled dot product attention (as per Eq. 5.27) is applied to each branch (Eq. 5.28).

$$\mathcal{B}_i = \alpha_{i \in [1,n]}(query_i\ \omega_i^{query}, key_i\ \omega_i^{key}, value_i\ \omega_i^{value}) \tag{5.28}$$

Afterwards, a residual connection is utilized on these tensors and a batch normalization layer is applied layer-wise, subsequently. Then, a scaling factor μ is employed to generate the branch representation as follows:

$$\tilde{\mathcal{B}}_i = \texttt{LayerNorm}(\mathcal{B}_i \omega_i^b + \mathcal{B}_i) \times \mu_i \tag{5.29}$$

Subsequently, a position-wise CNN (PCNN) is employed to every $\tilde{\mathcal{B}}_i$. This PCNN layer consists of two convolution operations on each position with a `ReLU` activation function in between.

This PCNN layer works as Eq. 5.30:

$$\texttt{PCNN}(x) = \texttt{Conv}(\texttt{ReLU}(\texttt{Conv}(x) + b_1)) + b_2 \tag{5.30}$$

The final attentive representation of these semantic sub-spaces, generated from the PCNN layer, is obtained by performing a linear weighted summation (Eq. 5.31) where $\gamma \in \mathbb{R}^n$ is a model hyper-parameter.

$$\texttt{BranchAttn} = \sum_{i=1}^{n} \gamma_i \texttt{PCNN}(\tilde{\mathcal{B}}_i) \tag{5.31}$$

In the last step, a residual connection is established with `BranchAttn` and non-linearity (tanh) is applied. The parent node representation is achieved by performing element-wise summation (`ExS`). Eq. 5.32 represents the operation of this step.

$$\texttt{ParentNode} = \texttt{EWS}(\tanh((\chi_{attn} + \chi)\omega + b)) \tag{5.32}$$

In Eq. 5.32, χ and χ_{attn} symbolize the input and output features of the attention computation module.

5.2.3.2 Heterogeneous Graph Attention Network

The heterogeneous graph attention network (H-GAT) [227] was initially introduced for the textual summarization task to provide enriched cross-sentence relationships. In this work, we have utilized this approach to improve the sentence representation quality. At each iteration, this module is deployed once the constituency and dependency tree-transformers' forward passes are done. Via sentence-to-word and word-to-sentence update processes, this module provides enriched sentence vectors.

For this module the graph G has been structured as $G = \{V, E\}$. The set V represents the nodes in the graph, while E represents the edges between those nodes. For any sentence S containing n words (w_i), $V = \{w_1, w_2, ..., w_n, S\}$. As this task finds protein-protein interactions in single sentences, the edges are established in such a way that the sentence node S is connected to every word node w_i. Once the graph G has been constructed, a Graph Attention Network (GAT) [224] is employed to modify the feature values of the nodes. Let $h_i \in \mathbb{R}^{d_h}$ be the hidden states of the word and sentence nodes, where $i \in \{1 : (n + 1)\}$ and d_h is the hidden state

dimension. Then the GAT layer can be represented as:

$$\kappa_{i,j} = \texttt{LeakyReLU}(\omega_a[\omega_q h_i; \omega_k h_j]) \tag{5.33}$$

$$\alpha_{i,j} = \frac{exp(\kappa_{i,j})}{\sum_{l \in \mathcal{N}_i} exp(\kappa_{i,l})} \tag{5.34}$$

$$\mathcal{Z}_i = \sigma(\sum_{j \in \mathcal{N}_i} \alpha_{i,j} \omega_v h_j) \tag{5.35}$$

where the ω_a, ω_q, ω_k, and ω_v weight-matrices are updated via backpropagation. The set of neighbouring nodes for any considered node is represented by $\mathcal{N}_i$. The attention score between h_i and h_j is represented by $\alpha_{i,j}$. The GAT incorporating multi-head attention, with $\mathcal{M}$ attention heads, can be defined as:

$$\mathcal{Z}^i = \|_{m=1}^{\mathcal{M}} \sigma(\sum_{j \in \mathcal{N}_i} \alpha_{i,j}^m \omega^m h_i) \tag{5.36}$$

In order to avoid the vanishing of gradients over time, a residual connection is also established. With the information u_i from this residual connection, the final hidden state representation is formulated as follows:

$$h_i = u_i + h_i \tag{5.37}$$

By means of the previously described GAT and a position-wise feed forward network (FFN) layer, which consists of two linear transformations [227], the word nodes are updated based on the information from the sentence node seen in Eqs. 5.38 and 5.39:

$$\mathcal{Z}_{s \to w}^1 = GAT(\mathcal{H}_w^0, \mathcal{H}_s^0, \mathcal{H}_s^0) \tag{5.38}$$

$$\mathcal{H}_w^1 = FFN(\mathcal{Z}_{s \to w}^1 + \mathcal{H}_s^0) \tag{5.39}$$

where $\mathcal{H}_w^0$ is the set of word nodes (the Bio-RoBERTa-based embeddings for words [82]) for the words present in the sentence. $\mathcal{H}_s^0$ is the average of the sentence representations from the dependency and constituency tree-transformers. In Eq. 5.38, $\mathcal{H}_w^0$ has been considered as the query matrix and $\mathcal{H}_s^0$ has been considered as both the key and value matrices following the work of Vaswani et al. [223].

After updating the word nodes based on the sentence node, the next step involves updating the sentence node based on the just updated word nodes. These sentence-to-word and word-to-sentence node refinement processes continue at each iteration. For the t-th iteration, the process

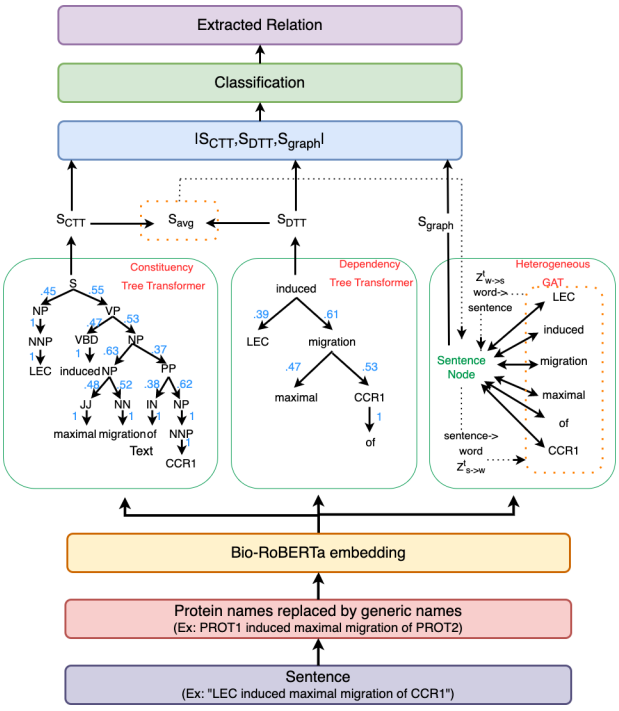

Figure 5.2: This diagram illustrates the approach for combining features from the dependency and constituency tree-transformers together with a heterogeneous graph attention network to create an integrated architecture for PPI prediction. The blue numbers in the Constituency and Dependency Tree Transformers indicate the attention value for the associated tree branches.

can be represented in the following manner:

$$\mathcal{Z}_{s\to w}^{t+1} = GAT(\mathcal{H}_w^t, \mathcal{H}_s^t, \mathcal{H}_s^t) \tag{5.40}$$

$$\mathcal{H}_w^{t+1} = FFN(\mathcal{Z}_{s\to w}^{t+1} + \mathcal{H}_w^t) \tag{5.41}$$

$$\mathcal{Z}_{w\to s}^{t+1} = GAT(\mathcal{H}_s^t, \mathcal{H}_w^{t+1}, \mathcal{H}_w^{t+1}) \tag{5.42}$$

$$\mathcal{H}_s^{t+1} = FFN(\mathcal{Z}_{w\to s}^{t+1} + \mathcal{H}_s^t) \tag{5.43}$$

5.2.3.3 Model Architecture

Figure 5.2 sketches the overall architecture of the model. Each unit of the model is initially fed with the Bio-RoBERTa [82] word embeddings. Then the constituency and dependency tree transformers generate the sentence representations (S_{CTT} and S_{DTT}, respectively) in parallel. A point-wise average operation is applied to these two sentence vectors. This averaged sentence vector (S_{avg}) is then used for the sentence-to-word and word-to-sentence update steps in the heterogeneous graph attention network. This step provides another sentence representation (S_{graph}). In the following step, max-pooling is applied and followed by a multi-layer perceptron for the PPI extraction.

5.2.4 Experimental Details and Performance Analysis

In this section, we present the performance of the proposed model, evaluated using the F1-score. We have formulated the PPI extraction as a classification task. We conclude by comparing the efficacy of the proposed model to the leading sequential, tree-structured, and graph-based architectures that have been previously proposed for the PPI task. We first include a statistical overview of the five primary PPI corpora utilized in this task, as well as a discussion of the pre-processing techniques employed on these corpora.

5.2.4.1 Corpus Description

First, to assess the performance of the examined model, we evaluate its performance on the five standard PPI benchmark corpora: BioInfer [173], AIMed [35], HPRD50 [69], IEPA [57], and LLL [158]. In all of the experiments, the following transformed version of each corpus is employed, as specified by Ahmed et al. [6] and Singha Roy and Mercer [204]. To provide a consistent classification task across all five corpora, all protein names are replaced with three distinct symbols: if a pair of proteins are to be considered as potentially interacting in a given sentence, they are substituted with the labels PROT1 and PROT2 and all other proteins mentioned in the sentence are substituted with PROT0. Thus, this approach has the model consider an interaction between two proteins, one at a time. To work with sentences containing more than two proteins, two proteins at a time are tagged with PROT1 and PROT2 and their interaction (positive or negative) is identified. Sequentially, all protein pairs are considered. So, for each sentence in the corpus containing η proteins, the modified corpus will feature ${}^{\eta}C_2$ variations. As an example, consider the following sentence: "At 89.3 nmol/L, maximal migration of CCR1 and CCR8 transfected cells was prompted by LEC and at 5.6 nmol/L, cell adhesion also occurred." To identify the possible relationship between LEC and CCR1, we replace their respective protein names with PROT1 and PROT2, while replacing CCR8 with PROT0. When the objective is to identify the possible interaction between LEC and CCR8, we replace their names with PROT1 and PROT2, and use PROT0 in place of CCR1. Similarly, when identifying the possible interaction between CCR1 and CCR8, they are replaced with PROT1 and PROT2 and LEC is replaced with PROT0. Interactions between protein pairs can be either positive or negative. For the above example, when the considered proteins are CCR1 and LEC or CCR8 and LEC, the nature of their interactions is positive in each case. However, when the considered protein pair is CCR1 and CCR8, the PPI is negative since no interaction is present between them. Thus the example sentence presents three possible interactions, resulting in three variants (${}^{3}C_2$) of the sentence in the modified corpus: two with positive interactions, and one with a negative interaction. Using generic names to represent protein names enhances

Table 5.3: Statistics of the modified corpora

Corpus	Original Sentences	Positive Interactions	Negative Interactions
AIMED	1,995	1,000	4,834
BioInfer	1,100	2,534	7,132
IEPA	486	335	482
HPRD50	145	163	270
LLL	77	164	166

the data by allowing for multiple samples of these generic names, as opposed to only a few samples for each individual protein name. Table 5.3 provides an overview of the demographic characteristics of the five modified corpora applying this above-mentioned approach. We have utilized the Stanford dependency and constituency parser to parse sentences in each corpus [137].

5.2.4.2 Experimental Setup

Next, turning to the details of the model, it uses an initial learning rate of 0.1. If the validation accuracy decreases from the previous iteration, the learning rate is reduced by 80% in each iteration. We set the batch size to 10. The tree-transformer models use six PCNN layers and six branches of attention layer, and employ 341-dimension and 300-dimension kernels in two CNN layers with dropout 0.1 in the second layer only. For the H-GAT unit, six attention heads are utilized. The model hyper-parameters are trained by the 'Adagrad' [132] optimizer. The final sentence representation of each individual unit as well as the model is a 512-dimensional vector. The model is fed with Bio-RoBERTa word embeddings. For the tree-transformers, these embeddings are not further updated. But, for the H-GAT unit, with the sentence-to-word update step, word embeddings are updated once each epoch. We have also experimented with PubMed-BERT [80], however, better performance is acquired when Bio-RoBERTa word embeddings are used. We have utilized StratifiedK-Fold from the scikit-learn package to perform 10-fold cross-validation in the model evaluation process. For each fold, the training was done on the training set and the test was done on a separate test set.

All of the experiments are performed on Linux Ubuntu 22.04 LTE with 16GB memory and Nvidia 1070Ti 8GB graphics memory. For implementing the model, we have used PyTorch 1.7.1. In this environment, the model took 8 hours each for training on the BioInfer and AIMed corpora.

Table 5.4: Performance evaluation of the models by means of F1-score (in %). The sequential, tree-structured, and graph-based models are tagged with †, ‡, and ∗, accordingly. The performance metric of our model is presented in **bold**.

Methods	AIMed	BioInfer	IEPA	HPRD50	LLL	Avg.
Chang et al. [42] †	60.6	69.4	71.4	71.5	80.6	70.7
Hsieh et al. [88] †	76.9	87.2	76.31	80.51	78.3	79.84
Zhang et al. [267] †	56.4	61.3	75.1	63.4	76.5	66.54
Yadav et al. [245] †	77.33	76.33	-	-	-	76.83
Tai et al. [214] ‡	80.6	88.1	76.4	82.0	84.8	82.38
Ahmed et al. [4] ‡	81.6	89.1	78.5	81.3	84.2	82.94
Singha Roy and Mercer[204] ‡	88.15	96.01	83.24	88.94	92.18	89.70
Fei et al. [64] ∗	88.27	96.21	83.90	89.57	92.86	90.16
Proposed Model	**91.23**	**96.97**	**87.28**	**93.11**	**93.52**	**92.02**

5.2.4.3 Performance Analysis

Table 5.4 displays how our proposed model performs on the five benchmark corpora, along with the published results of several sequential, tree-structured, and graph-based models for comparison. For performance evaluation, we have used the F1-score. With the AIMED corpus, we have achieved 91.23% F1-score, which is a 2.96 percentage point (p.p.) performance boost compared to the current state of the art [64]. The second dataset that has been used to evaluate the model is BioInfer. It has the highest number of annotated interactions compared to the other four datasets. In this dataset the sentences are notably longer and encompass a greater number of protein names mentioned within a single sentence. On this corpus, our model has achieved a F1-score of 96.97% which is 0.76 p.p. and 0.96 p.p. higher than Fei et al. [64] and Singha Roy and Mercer [204], respectively. The three remaining corpora (IEPA, HPRD50, and LLL) come with comparably smaller number of samples. Even for these corpora with very few samples our model has outperformed the current state of the art model [64] for the PPI task. Compared to Singha Roy and Mercer [204], which is the best performing tree-structured model for the PPI extraction task, our model has gained 4.04 p.p., 4.17 p.p., and 1.34 p.p. higher F1-scores for the IEPA, HPRD50, and LLL corpora, accordingly. In comparison to the work of Fei et al. [64], for these three corpora, in the order given above, the performance boosts for our model are 3.38 p.p., 3.54 p.p., and 0.66 p.p. On average, over these five corpora, our model has achieved 92.02% F1-score which is 1.86 p.p. higher than what is reported in Fei et al. [64].

Further to these performance numbers, it is noteworthy that if we discard the H-GAT module, our proposed model is almost identical to the model presented in the work of Singha Roy and Mercer [204]. Comparing the F1-scores of these two models in Table 5.4, we can see that the sentence-to-word and word-to-sentence update processes are key to the improvement

Table 5.5: Cross-corpus experimental results by means of F1-score (in %). The training corpora are represented by the rows, while the testing data is presented in the columns. Rows marked with † are the results from Ahmed et al. [4].

	AIMed	*BioInfer*	*IEPA*	*HPRD50*	*LLL*
AIMED †	-	47.0	38.6	41.5	34.6
BioInfer †	50.8	-	40.8	45.5	33.5
AIMED	-	55.1	42.7	46.2	39.5
BioInfer	56.7	-	44.0	50.3	40.8

seen in the performance of our new model. This enhanced performance, we believe, is because when the H-GAT module is being employed, the sentence representations generated by the tree-transformers, and thus the newly generated word representations by the sentence-to-word update step, provide a more enriched semantics for the task which in turn help to produce (due to the word-to-sentence process) a better sentence representation for the following classifier. Our belief is further supported by noting that the max-pooling layer, having replaced a sentence feature concatenation layer in a previous version of our model, results in a 0.5-1.8 p.p. performance boost compared to the previous version (previous model's numbers not shown).

5.2.4.4 Cross-Corpus Performance Analysis

In addition, we have performed a cross-corpus assessment, motivated by Van et al. [222], which aims to address a critical inquiry regarding the effective extraction of protein-protein interactions in practical applications – "which corpus is most suitable for the training of a specific model in real-world scenarios?". Table 5.5 shows the results achieved by our model for the cross-corpus evaluation. The training data is represented by the rows, and the test data is represented by the columns. In this study, we utilized AIMed and BioInfer exclusively as the training datasets while disregarding the smaller ones. This is because training on small and simple corpora and testing on larger, more intricate datasets serves no practical purpose [164, 4]. The results show a noticeable decline in performance across all corpora due to the lack of consistency between the distribution of the training and testing data. The acquired results support the basic principle of machine learning, which states that training and test sets should have identical distributions. Notably, our proposed model trained on BioInfer outperforms the same model trained on AIMed, likely due to the former's larger size. The results also show that, for our model, if it is used in real life scenarios, BioInfer should be the suggested corpus for model training. Furthermore, these transfer learning results show a performance boost compared with Ahmed et al. [4].

5.2.5 Conclusion and Future Work

In this paper, we have proposed a supervised Protein-protein interaction extractor model which has the ability to obtain the word level dependencies, phrasal information, and better semantics by means of utilizing dependency and constituency tree-transformers and a heterogeneous graph neural network. Our model has shown significant performance improvements on all five benchmark corpora.

Despite the progress made in this work, there is still room for further improvement. The sentence-to-word and word-to-sentence node update step can be applied directly over the tree-transformers to see how they perform. Additional analysis of the results can be conducted by examining the AUC and ROC curves.

5.3 Extracting Drug-Drug and Protein-Protein Interactions from Text Using a Continuous Update of Tree-Transformers

This chapter is based on the paper titled "Extracting Drug-Drug and Protein-Protein Interactions from Text Using a Continuous Update of Tree-Transformers" co-authored with Robert E. Mercer that appeared in the *Proceedings of the 22nd Workshop on Biomedical Language Processing (BioNLP 2023)* [205].

Understanding biological mechanisms requires determining mutual protein-protein interactions (PPI). Obtaining drug-drug interactions (DDI) from scientific articles provides important information about drugs. Extracting such medical entity interactions from biomedical articles is challenging due to complex sentence structures. To address this issue, our proposed model utilizes tree-transformers to generate the sentence representation first, and then a sentence-to-word update step to fine-tune the word embeddings which are again used by the tree-transformers to generate enriched sentence representations. Using the tree-transformers helps the model preserve syntactical information and provide semantic information. The fine-tuning provided by the continuous update step adds improved semantics to the representation of each sentence. Our model outperforms other prominent models with a significant performance boost on the five standard PPI corpora and a performance boost on the one benchmark DDI corpus that are used in our experiments.

5.3.1 Introduction

With the rapid expansion of scientific literature, most biological knowledge is now stored as text and can be accessed through scientific publications. The MEDLINE database has experienced a steady annual growth of over 4% for the last two decades, currently boasting a collection of over 29 million records from diverse sources. This is an increase of 3 million records compared to 2020 and over 8 million records compared to 2014, as cited in Yadav et al. [245]. The vast amount of textual data in biomedical research articles presents an invaluable opportunity for automated biomedical information retrieval to leverage this wealth of information.

As biomedical data continues to expand exponentially and the inherent complexity of textual representations, automated methods for information retrieval plays a pivotal role in aiding biologists in locating pertinent information, managing databases, and providing decision support to medical practitioners. Numerous studies have been conducted to extract valuable information from these texts, encompassing various domains such as protein-protein interactions, chemical-disease relationships, clinical correlations, drug-drug interactions, and more.

The internal biological processes within a cell, such as cellular organization, signal transduction, and immune response, are predominantly governed by interactions between different proteins [209]. To comprehend the molecular mechanisms underlying these biological processes, knowledge of protein-protein interactions (PPI) is indispensable [4]. These interactions have significant relevance in the biomedical domain, including drug target examination [77] and signal proteins [9]. Consequently, the identification of protein-protein interactions (PPIs) leads to a deeper understanding of the functions, regulation, and communication between various proteins [250]. The primary objective of PPI recognition is to extract the relationships between protein entities mentioned in a document [112].

A drug-drug interaction (DDI) refers to a modification in the effects of one drug due to the presence of another drug [186]. While clinical trials for pre-market identification of interactions are challenging, obtaining DDI information from scientific articles is a faster, cost-effective, and reliable approach to reducing adverse effects. Furthermore, in order to practice evidence-based medicine and mitigate drug-related accidents, comprehensive extraction of DDI knowledge from pharmaceutical literature is crucial [190]. Automatic DDI extraction can prove highly beneficial for the pharmaceutical industry, offering an efficient means of reducing the time spent by healthcare professionals reviewing the medical literature.

Biomedical literature contains a wealth of information about protein-protein interactions (PPIs) and drug-drug interactions (DDIs), but this information is often unstructured. Manual extraction of these interactions from biomedical literature is a laborious, resource-intensive,

and costly task, given the sheer volume of published studies [165, 216]. Consequently, the automatic extraction of PPIs and DDIs from biomedical literature has emerged as a vital research area, garnering attention from numerous researchers. While the information may be scattered throughout the document, the current study focuses on detecting these interactions within individual sentences, similar to previous studies [15, 64, 4, 218].

An instance of a sentence that demonstrates protein-protein interactions can be found in the study by Howard et al. [87], where it states:

> "At 89.3 nmol/L, maximal migration of CCR1 and CCR8 transfected cells was prompted by LEC and at 5.6 nmol/L, cell adhesion also occurred."

This sentence highlights two protein-protein interactions involving LEC and CCR1, as well as LEC and CCR8. However, it is important to note that there is no correlation mentioned between proteins CCR1 and CCR8 in this context.

An instance of a DDI-containing sentence is [157]:

> "To determine whether probenecid has a direct effect on the distribution of cloxacillin, the elimination and distribution of cloxacillin was studied in six patients, five lacking kidney function and one with a partially impaired renal function, in the presence or absence of probenecid."

This sentence mentions two drugs: probenecid, and cloxacillin. However, the interaction between them is negative, as no concrete interaction is stated.

During the extraction of relationships between target proteins or drugs, we have addressed three key concerns. Firstly, how to tackle the challenge of retrieving relations when the mentioned proteins or drugs are widely separated in the text. Secondly, how to preserve better semantics by handling the phrasal structure of the text, allowing for the effective extraction of PPIs or DDIs and capturing relevant information. Lastly, what is the impact of updating word and non-leaf node representations in the tree-structured networks based on the sentence at hand, as opposed to using fixed representations from pre-trained models, and how this influences the generated sentence representation for the task of PPI and DDI extractions.

To address the above-mentioned three considerations we have proposed a model combining a constituency tree-transformer (for preserving phrase-level information in the text), and a dependency tree-transformer (to consider relations between long distant drugs or proteins in the text) where each of them generates sentence representations which are then combined. Finally, a sentence-to-word update step is introduced following the concept from Wang et al. [227] to update the word and non-leaf nodes of the tree-transformers to generate refined sentence representation. This approach serves the purpose of fine-tuning BERT-based word embeddings for

these tasks. But the advantage of this approach is that we do not need to fine-tune millions of parameters in the BERT-based models. Our study includes a thorough analysis of the performance of the proposed models on benchmark PPI and DDI datasets. The results demonstrate the superiority of our proposed model compared to previous prominent models in the field. The comprehensive analysis highlights the effectiveness and efficacy of our approach in accurately extracting protein-protein and drug-drug interactions from biomedical literature.

5.3.2 Related Work

In the initial stages of biomedical entity relation extraction research, co-occurrence and pattern recognition techniques were commonly employed [20, 256]. However, with advancements in technology, machine learning techniques have gained prominence due to their superior performance. Early approaches involved feature engineering and kernel methods to construct a feature set, followed by classification using support vector machines or other classifiers [7, 152]. In recent years, deep learning techniques, leveraging the widespread use of deep learning in natural language processing (NLP), have been successfully applied to PPI and DDI extraction in several research works [125, 270, 89, 49]. Zhang et al. [267] proposed a three-channel convolution neural network for extracting PPIs from the text.

Recent work in PPI and DDI extraction often utilizes recurrent neural network (RNN) models that treat textual representations as sequences [88, 192, 244]. However, these models may miss semantic compositions when biomedical entities lie at distant positions in the text, as they only consider word order and ignore linguistic structure [5, 118]. In contrast, recursive neural networks, also known as tree-structured neural network models [6, 204], process sentences represented in a parsed tree form, capturing both syntax and semantics in a more effective manner. There have also been investigations into graph-based methods, where models operate on a fully connected graph composed of either word or phrase nodes [64]. These approaches aim to leverage the structural information present in the data for improved performance in PPI and DDI extraction. Asada et al. [14] utilized molecular structure and description of the drugs for retrieving DDIs. Gu et al. [80] fine-tuned PubMedBERT to extract relations between drugs. Following this, Asada et al. [15] utilized a knowledge graph with PubMedBERT for the DDI extraction task.

5.3.3 Proposed Model

In this section, we provide details of our model for the protein-protein and drug-drug interaction extraction tasks. Our model contains three key modules: two tree-transformers, as described in Ahmed et al. [6], for preserving the semantic and syntactical information, and a sentence-

to-word update step for updating the word and intermediate node representations in the tree-transformers to generate refined representations of the sentences. In this current work, we have added an update of the word embeddings after the sentence-to-word update step which enriches the input to the combination of the two tree-transformers and the heterogeneous graph attention network, which were first proposed for the PPI extraction task in Singha Roy and Mercer [201]. In this section, we first discuss how each module functions individually, and then elaborate on how these modules are integrated into our proposed model with the expanded workflow.

5.3.3.1 Tree-Transformers

The two tree-based representations commonly used for representing a sentence are constituency trees and dependency trees. Constituency trees capture the structure of phrases in a sentence, while dependency trees represent the dependencies between individual words. In our work, we utilize two tree-transformer models, namely the dependency tree-transformer and the constituency tree-transformer, as proposed by Ahmed et al. [6], to leverage these sources of syntactic structure information. The goal of these tree-transformer models is to traverse each sub-tree within a dependency or constituency tree structure attentively and at its root derive a sentence representation. This allows us to capture both the semantic and syntactic information of the sentence for improved performance in extracting protein-protein and drug-drug interactions from the text. Unlike the tree transformer proposed by Wang et al. [229] which learns phrases they call constituents, the tree transformer proposed by Ahmed et al. [6] works over the parsed trees and can work with both the constituency and dependency trees.

In a dependency tree, each node represents a word in the sentence. When traversing a sub-tree in a dependency tree, the dependency tree-transformer takes into consideration the representations of both the parent and child nodes, allowing for the propagation of information between connected words in the tree. On the other hand, in a constituency tree, only the leaf nodes hold words, while the non-terminal nodes do not have word representations. The vector representations for the non-terminal nodes are computed only after the sub-tree has been fully traversed, taking into account the information from the leaf nodes. This approach allows for the capture of both local and global contextual information during the tree traversal process, facilitating the extraction of meaningful representations from the syntactic structures of the sentence. Ahmed et al. [6] have used a self-attention mechanism to process the dependency and constituency tree representations of the sentence, employing query ($\mathcal{Q}$), key ($\mathcal{K}$), and value ($\mathcal{V}$) matrices, which are computed as follows based on the formulation proposed by Vaswani

et al. [223]:

$$\mathcal{K} = \omega_k \mathcal{M}_k \quad \text{s.t.} \quad \omega_k \in \mathbb{R}^{d \times d} \tag{5.44}$$

$$\mathcal{V} = \omega_v \mathcal{M}_v \quad \text{s.t.} \quad \omega_v \in \mathbb{R}^{d \times d} \tag{5.45}$$

$$\mathcal{Q} = \omega_q \mathcal{M}_q \quad \text{s.t.} \quad \omega_q \in \mathbb{R}^{d \times d} \tag{5.46}$$

In the tree-based transformer models, the matrix $\mathcal{M}$ is computed differently for dependency trees and constituency trees. In the case of dependency trees, the matrix $\mathcal{M}$ is created by concatenating the word vectors of all of the child nodes for each parent node in the dependency tree. On the other hand, for constituency trees, $\mathcal{M}$ is formed by concatenating the word vectors within a constituent. The self-attention matrix α is then computed as:

$$\alpha = \text{softmax}(\frac{\mathcal{Q}\,\mathcal{K}^T}{\sqrt{d_k}})\mathcal{V} \tag{5.47}$$

where d_k represents the dimension of $\mathcal{K}$. To implement multi-branch attention with n branches, the following steps are taken: first, n copies of the key, query, and value matrices are generated using weight matrices (ω_i). Then, each branch applies the scaled dot product attention separately (following Eq. 5.47), using its own set of query, key, and value vectors. Finally, this results in n sets of attended word vectors, one for each branch (see Eq. 5.48).

$$\mathcal{B}_i = \alpha_{i \in [1,n]}(\mathcal{Q}_i\ \omega_i^{\mathcal{Q}}, \mathcal{K}_i\ \omega_i^{\mathcal{K}}, \mathcal{V}_i\ \omega_i^{\mathcal{V}}) \tag{5.48}$$

Then, a residual connection is employed on these tensors, and a batch normalization layer is applied to each layer. Following that, the branch representation is generated using a scaling factor μ in the following manner:

$$\tilde{\mathcal{B}}_i = \texttt{LayerNorm}(\mathcal{B}_i \omega_i^b + \mathcal{B}_i) \times \mu_i \tag{5.49}$$

Following that, a position-wise CNN (PCNN) is applied to each $\tilde{\mathcal{B}}_i$. The PCNN layer comprises two convolution operations on each position, separated by a rectified linear unit (ReLU) activation function. The operation of this PCNN layer can be represented as per Eq. 5.50:

$$\texttt{PCNN}(x) = \texttt{Conv}(\texttt{ReLU}(\texttt{Conv}(x) + b_1)) + b_2 \tag{5.50}$$

The ultimate attentive representation of these semantic sub-spaces, which are generated from the PCNN layer, is acquired by conducting a linear weighted summation (as expressed in Eq.

5.51), with $\gamma \in \mathbb{R}^n$ serving as a hyper-parameter of the model.

$$\texttt{BranchAttn} = \sum_{i=1}^{n} \gamma_i \texttt{PCNN}(\tilde{\mathcal{B}}_i) \tag{5.51}$$

In the final stage, a residual connection is established with `BranchAttn`, and a hyperbolic tangent non-linearity (tanh) function is applied. The representation of the parent node is then obtained by conducting element-wise summation (`EWS`) (Eq. 5.52).

$$\texttt{ParentNode} = \texttt{EWS}(\tanh((\chi_{attn} + \chi)\omega + b)) \tag{5.52}$$

In Eq. 5.52, the symbols χ and χ_{attn} represent the input and output features of the attention computation module, respectively.

5.3.3.2 Sentence-to-Word Update Module

For the sentence-to-word update step, we have used an approach similar to the heterogeneous graph attention network (H-GAT) [227]. H-GAT was introduced for extractive summarization tasks with the intention to generate an enriched cross-sentence relationship. In our research, we have employed this approach to enhance the quality of sentence representations. This module is utilized at each iteration, once the forward passes of the constituency and dependency tree-transformers are completed. Through sentence-to-word and a following forward pass of the tree-transformers again, this module enriches the sentence vectors, thereby improving the overall sentence representation quality.

The graph G in this module is structured as $G = V, E$, where V represents the nodes in the graph and E represents the edges between those nodes. For a given sentence S containing n words (w_i), the set of nodes V is defined as $V = w_1, w_2, ..., w_n, S$. Since the task involves identifying PPIs and DDIs in single sentences, the edges are established in such a way that the sentence node S is connected to every word node w_i. Once the graph G is constructed, a Graph Attention Network (GAT) [224] is used to modify the feature values of the nodes. Let $h_i \in \mathbb{R}^{d_h}$ be the hidden states of the word and sentence nodes, where $i \in 1 : (n + 1)$ and d_h is the hidden state dimension. The GAT layer can be formulated as follows:

$$\kappa_{i,j} = \texttt{LeakyReLU}(\omega_a[\omega_q h_i; \omega_k h_j]) \tag{5.53}$$

$$\alpha_{i,j} = \frac{\exp(\kappa_{i,j})}{\sum_{l \in \mathcal{N}_i} \exp(\kappa_{i,l})} \tag{5.54}$$

$$\mathcal{Z}_i = \sigma(\sum_{j \in \mathcal{N}_i} \alpha_{i,j} \omega_v h_j) \tag{5.55}$$

The weight matrices ω_a, ω_q, ω_k, and ω_v in the GAT layer are updated through backpropagation. The set of neighbouring nodes for a given node i is denoted by $\mathcal{N}i$, while the attention score between hidden states h_i and h_j is denoted by $\alpha_{i,j}$. The GAT layer can be extended to incorporate multi-head attention with $\mathcal{M}$ heads, which is represented as follows:

$$\mathcal{Z}^i = \|_{m=1}^{\mathcal{M}} \sigma(\sum_{j \in \mathcal{N}_i} \alpha_{i,j}^m \omega^m h_i) \tag{5.56}$$

To mitigate the issue of vanishing gradients over time, a residual connection is established. The final hidden state representation (h_i), incorporating the information (u_i) from the residual connection, is formulated as $h_i = u_i + h_i$.

The word nodes are updated using the previously delineated GAT and a position-wise feed-forward network (FFN) layer, which consists of two linear transformations as introduced by Wang et al. [227]. At the t-th iteration, the updates are performed based on the information from the sentence node, as shown in Eqs. 5.57 and 5.58:

$$\mathcal{Z}_{s \to w}^{t+1} = GAT(\mathcal{H}_w^t, \mathcal{H}_s^t, \mathcal{H}_s^t) \tag{5.57}$$

$$\mathcal{H}_w^{t+1} = FFN(\mathcal{Z}_{s \to w}^{t+1} + \mathcal{H}_s^t) \tag{5.58}$$

In Eq. 5.57, $\mathcal{H}_w^0$ represents the set of word nodes, which are the Bio-RoBERTa-based embeddings for the words in the sentence [82]. On the other hand, $\mathcal{H}_s^t$ represents the average of the sentence representations obtained from the dependency and constituency tree-transformers. In the GAT layer, $\mathcal{H}_w^t$ is used as the query, while $\mathcal{H}_s^t$ is considered as both the value and key matrices, imitating the approach of Vaswani et al. [223].

5.3.3.3 Model Architecture

Figure 5.3 provides an architectural overview of the model. The model starts with Bio-RoBERTa word embeddings as input. These embeddings are then processed by the Dependency Tree Transformer (DTT) and Constituency Tree Transformer (CTT) in parallel to generate sentence representations (S_{DTT} and S_{CTT}, accordingly). This step is followed by a mean-pooling operation and an intermediate sentence representation S_{avg} is generated. The sentence-to-word update step uses the S_{avg} representation to update the word representations. These updated word representations are then passed to the tree-transformers again. This step involves another forward pass to generate the updated sentence representations S'_{DTT} and S'_{CTT}. Max-pooling is applied over these updated sentence representations and this result is fed to the following classification layer for the relation extraction.

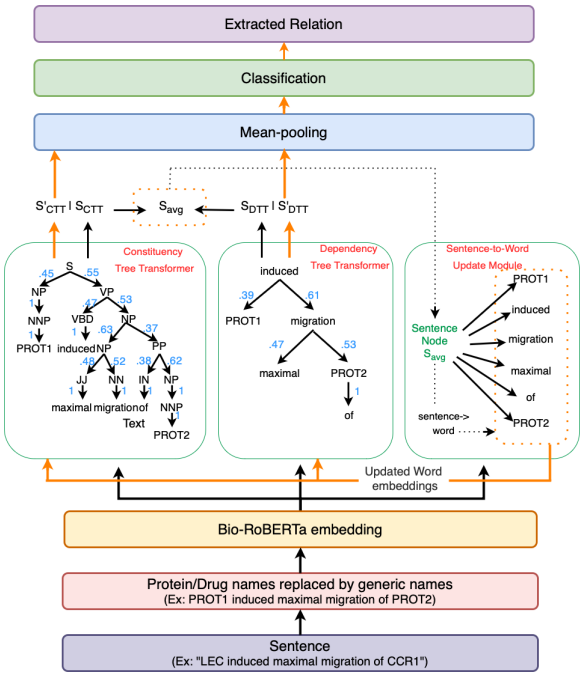

Figure 5.3: Integrated architecture with tree-transformers with the sentence-to-word update step for relation extraction task. The numerical values in blue color, associated with the branches in the tree Transformers, represent the attention scores for those specific branches.

5.3.4 Experimental Setup and Analysis of Results

In this section, the performance of the proposed model is evaluated using the F1-score. The PPI (Protein-Protein Interaction) and DDI (Drug-Drug Interaction) extraction tasks have been formulated as classification tasks. The section also includes a demographical overview of the five primary PPI corpora and the standard DDI corpus used in the evaluation, as well as a discussion of the pre-processing techniques employed on these corpora. The efficacy of the proposed model is compared to leading sequential, tree-structured, and graph-based architectures that have been previously propounded for these biomedical entity inter-relation extraction tasks.

5.3.4.1 Corpora Descriptions

The performance of the proposed model for PPI extraction task is evaluated on five benchmark corpora: BioInfer [173], AIMed [35], HPRD50 [69], IEPA [57], and LLL [158]. In order to bring forth a persistent classification task across all five corpora, protein names are substituted with three symbols: PROTEIN1 and PROTEIN2 are used to represent pairs of proteins that are considered potentially interacting in a given sentence, while all other protein names present in the sentence are altered with PROTEIN0. The approach of replacing protein names with

Table 5.6: Demographical description of the modified corpora for PPI task

Corpus	Original Sentences	Positive Samples	Negative Samples
AIMED	1,995	1,000	4,834
BioInfer	1,100	2,534	7,132
IEPA	486	335	482
HPRD50	145	163	270
LLL	77	164	166

generic symbols allows the model to focus on the interaction between a pair of proteins in each sentence, one at a time. For sentences containing more than two proteins, two proteins at a time are tagged with PROTEIN1 and PROTEIN2, and their interaction (positive or negative) is identified. This process is repeated sequentially for all protein pairs in the sentence. Thus, for each sentence in the corpus containing η proteins, the modified corpus will feature ${}^{\eta}C_2$ variations. For example, consider the sentence: "At 89.3 nmol/L, maximal migration of CCR1 and CCR8 transfected cells was prompted by LEC and at 5.6 nmol/L, cell adhesion also occurred." To identify the possible relationship between LEC and CCR1, their respective protein names are replaced with PROTEIN1 and PROTEIN2, while CCR8 is replaced with PROTEIN0. When the objective is to identify the possible interaction between LEC and CCR8, their names are replaced with PROTEIN1 and PROTEIN2, and PROTEIN0 is used in place of CCR1. Similarly, when identifying the possible interaction between CCR1 and CCR8, they are replaced with PROTEIN1 and PROTEIN2, and LEC is replaced with PROTEIN0. Interactions between protein pairs can be either positive or negative. For the above example, when the considered proteins are CCR1 and LEC or CCR8 and LEC, the nature of their interactions is positive in each case. However, when the considered protein pair is CCR1 and CCR8, the PPI is negative since no interaction is present between them. Thus, the example sentence presents three possible interactions, resulting in three variants (${}^{3}C_2$) of the sentence in the modified corpus: two with positive interactions and one with a negative interaction. Using generic names to represent protein names enhances the data by allowing for multiple samples of these generic names, as opposed to only a few samples for each individual protein name. An overview of the demographic traits for the five revised datasets, using the aforementioned method, is presented in Table 5.6.

For the DDI extraction task, we have conducted our experiments on the DDIExtraction-2013 corpus [199]. For the data preprocessing step, the aforementioned steps have been similarly followed. Here, the potentially interacting drug pairs are replaced with DRUG1 and DRUG2 and the remaining drug names in the sentence are replaced with DRUG0. Thus, each sample considers the interaction between one pair of drugs at a time, similar to the PPI data

Table 5.7: Demographical description of the SemEval-2013 DDIExtraction task dataset

	Train	Test
Sentences	6976	1299
Drug Pairs	27792	5716
Positive Pairs	4021	979
Mechanism	1319	302
Effect	1687	360
Advice	826	221
Interaction	189	96
Negative Pairs	23771	4737

preprocessing step. The overall demographic of the corpus is presented in Table 5.7.

The Stanford dependency and constituency parsers [137] have been employed to parse sentences in all of these corpora.

5.3.4.2 Experimental Setup

Regarding the model specifics, an initial learning rate of 0.1 has been employed for all the experiments. If the validation accuracy declines compared to the previous iteration, the learning rate has been decreased by 80% in each subsequent iteration. Additionally, a batch size of 10 is set.

The tree-transformer models incorporate six branches of an attention layer and six PCNN layers. Two CNN layers utilize kernels of dimensions 341 and 300, respectively, with a dropout of 0.1 in the second layer only. The sentence-to-word update module employs six attention heads. The trainable hyperparameters of the model are updated using the Adagrad optimizer [132]. The final representation for each sentence representation unit (dependency and constituency tree-transformers) and the model itself is a 512-dimensional vector. Bio-RoBERTa word embeddings are used as the initial input of the model. The model uses two forward passes for sentence vector generation. Only the first forward pass uses these Bio-RoBERTa word embeddings. The second pass utilizes the updated word representations obtained from the sentence-to-word update module, as described in Section 5.3.3.3.

To conduct the performance evaluation of the Proposed Model for the PPI extraction task, we have employed StratifiedK-Fold from the scikit-learn package to perform 10-fold cross-validation. In each fold, the training has been carried out on the training set, and the evaluation has been performed on a separate test set. The tree LSTM proposed by Tai et al. [214] [1] has

[1]This model was not developed in particular for the PPI task. We were interested in its performance on this

Table 5.8: Performance evaluation of the models for PPI extraction on the five datasets: F1-score (in %) as the metric. All values, except for Tai et al. [214] and the Proposed Model, are those reported in the original works. The best performance metric for each dataset is indicated in **bold**.

Methods	Architecture	AIMed	BioInfer	IEPA	HPRD50	LLL	Avg.
Chang et al. [42]	RNN	60.6	69.4	71.4	71.5	80.6	70.7
Hsieh et al. [88]	RNN	76.9	87.2	76.31	80.51	78.3	79.84
Zhang et al. [267]	RNN	56.4	61.3	75.1	63.4	76.5	66.54
Yadav et al. [245]	RNN	77.33	76.33	-	-	-	76.83
Tai et al. [214]	Tree-structured	80.6	88.1	76.4	82.0	84.8	82.38
Ahmed et al. [4]	Tree-structured	81.6	89.1	78.5	81.3	84.2	82.94
Singha Roy and Mercer[204]	Tree-structured	88.15	96.01	83.24	88.94	92.18	89.70
Fei et al. [64]	Graph-based	88.27	96.21	83.90	89.57	92.86	90.16
Singha Roy and Mercer[201]	Tree-structured + Heterogeneous Graph	91.23	96.97	87.28	93.11	93.52	92.02
Proposed Model	Tree-structured + Heterogeneous Graph	**94.66**	**97.81**	**93.47**	**94.01**	**94.14**	**94.82**

been trained and tested by us following the aforementioned approach. All the other models' results are reported directly from their corresponding publications. For the DDI extraction task, the training and test sets have been shuffled 5 times using StratifiedK-Fold from the scikit-learn package to perform 5-fold cross-validation. The average performance metrics for both tasks are presented in Tables 5.8 and 5.9, respectively, and discussed in Section 5.3.4.3.

The experiments have been conducted on a Linux Ubuntu 22.04 LTE machine equipped with 16GB of memory and an Nvidia 1070Ti graphics card with 8GB of graphics memory. PyTorch 1.7.1 has been utilized for implementing the model.

5.3.4.3 Performance Analysis

Table 5.9: Performance evaluation of the models on SemEval-2013 DDIExtraction: **P**recision, **R**ecall, and **F1**-score (in %) as the metrics. All values, except for the Proposed Model, are those reported in the original works. The best performance metrics are indicated in **bold**.

Methods	Architecture	P	R	F1
Yadav et al. [244]	RNN	76.5	69.0	72.6
Gu et al. [80]	PubMedBERT	-	-	82.4
Phan et al. [169]	RNN	-	-	83.7
Asada et al. [14]	Knowledge-based	85.4	82.8	84.1
Asada et al. [15]	PubMedBERT + Knowledge	85.3	85.5	85.4
Fei et al. [64]	Graph-based	94.9	92.0	93.4
Proposed Model	Tree-structured + Heterogeneous Graph	**95.5**	**94.9**	**95.2**

task.

Table 5.10: Performance of the model on individual DDI types of the SemEval-2013 DDIExtraction dataset

Metric	Mech.	Effect	Advice	Interac.
P	95.83	96.77	95.10	94.33
R	94.27	95.64	94.89	94.61
F1	95.04	96.20	94.99	94.47

Table 5.11: The ablation study of the Proposed Model on the PPI and DDI corpora. All values are F1-scores.

Discarded Component	AIMed	BioInfer	IEPA	HPRD50	LLL	DDI
Constituency Tree-Transformer	89.32	95.66	85.82	90.46	92.01	91.63
Dependency Tree-Transformer	89.11	95.43	84.60	89.72	91.78	90.96
Sentence-to-Word Update Module	88.11	95.89	83.17	88.85	92.10	89.98

Table 5.8 showcases the performance of our proposed model on the five benchmark corpora for PPI extraction, along with the published results of various sequential, tree-structured, and graph-based models for comparison. The F1-score has been utilized as the performance evaluation metric.

Our proposed model has demonstrated outstanding performance on all benchmark corpora, particularly on the AIMED, IEPA and BioInfer datasets. For the AIMED corpus, our model has achieved an impressive F1-score of 94.66%, surpassing the current state-of-the-art (SOTA) model [64] by 6.39 percentage points (p.p.). For the BioInfer dataset, which has longer sentences and more protein names mentioned in a single sentence, our model has shown remarkable results achieving an F1-score of 97.81%, surpassing the SOTA and Singha Roy and Mercer[204] results by 1.6 p.p. and 1.8 p.p., respectively. Even for the IEPA, HPRD50, and LLL corpora, which have smaller sample sizes, our model has outperformed the current SOTA. Compared to the best performing tree-structured model [204], our model has achieved significant improvements of 10.23 p.p., 5.07 p.p., and 1.96 p.p. for the IEPA, HPRD50, and LLL corpora, respectively. In comparison to Fei et al. [64], our model has achieved performance boosts of 9.57 p.p., 4.44 p.p., and 1.96 p.p. for the same three corpora, respectively. On average, across all five corpora, our model has obtained an impressive F1-score of 94.82%, surpassing the results reported in Fei et al. [64] by 4.66 p.p.

Table 5.9 shows the precision (P), recall (R) and F1-score achieved by the proposed model

for the DDI extraction task over the SemEval-2013 DDIExtraction corpus along with previous prominent models and Table 5.10 portrays the performance of the model over each individual class of the corpus. From Table 5.9 it is clearly visible that the proposed model has outperformed the current SOTA [64] with a significant margin of 1.8 p.p. by achieving 95.2% F1-score. For each individual type, the model has achieved more than 94% F1-score which also proves the generalization capability of the proposed model.

The first attempt to extract PPIs from text incorporating a tree structured neural network model was by Ahmed et al. [4]. They have applied structured attention over tree-LSTMs and achieved an average of 82.94% F1-score over the 5 benchmark PPI corpora. Later, in our following work [204], we have applied tree-transformers and gained a 6.86 p.p. performance boost on average. This model almost reached Fei et al. [64]'s work which was the state-of-the-art at that time. In the next step, we have experimented with adding an heterogeneous graph attention network model [201] with the tree transformers and observed a further performance gain of 2.32 p.p. In the work reported here, we have utilized the same heterogeneous graph attention network to update the word embeddings to generate a refined sentence vector which has given us another 2.8 p.p. performance gain over the PPI corpora, giving a total improvement of 5.12 p.p. from our initial tree-transformer model [204]. In this present work we have also experimented with the DDI corpus to show the generalizability of the method and gained a 1.8 p.p. F1-score improvement over the previous state-of-the-art [64].

5.3.4.4 Ablation Study

To indicate the importance of each module in the Proposed Model, an ablation study has been performed and the results are presented in Table 5.11.

If the sentence-to-word update module is discarded the model is similar to the work of Singha Roy and Mercer[204] and we can see a significant drop in the F1-score when this module is discarded. For the five PPI extraction corpora, this F1-score drop is 5.12 p.p. on average. For the SemEval-2013 DDIExtraction dataset (mentioned as DDI in Table 5.11), this F1-score drop is 5.1 p.p. which reflects the effectiveness of the sentence-to-word update module. This process plays a critical role in capturing relevant contextual information from both the sentence and word levels, leading to the enhanced model performance.

We believe that the improved performance is due to the sentence-to-word update module leveraging the sentence representations generated by the tree-transformers fed with task-specific and context-enriched word vectors. These sentence representations, along with the newly generated word representations through the sentence-to-word update step, enrich the semantics of the task. Consequently, the second forward pass produces a more informative sentence representation for the subsequent classifier, contributing to the enhanced performance

of our model.

The significance of the sentence-to-word update module is also supported by the other two ablation experiments presented in the table. When only one of the tree-transformers is utilized with the sentence-to-word update module, it performs better than that individual tree-transformer for these tasks. As reported in Singha Roy and Mercer[204], the dependency tree-transformer achieves 89.06% F1-score over the PPI extraction corpora on average, where with the sentence-to-word update module it is 90.65%. In the case of the constituency tree-transformer, the performance boost is 1.33 p.p. A similar observation has been found for the experiments with the DDI corpus, as well.

5.3.5 Conclusions and Future Work

From these results and discussions in the previous sections, we can conclude that our model performs significantly better than the other prominent models even without using any additional features. The tree-transformers enable the proposed model to capture better semantics along with syntactical information. Additionally, the sentence-to-word update module provides more task-specific context-aware information, generating enriched word embeddings that further enhance the sentence representations for the PPI and DDI extraction tasks.

Although the model has achieved a significantly improved performance over the previous models, still there is scope for further improvement. Including a knowledge-graph, like in Asada et al. [15], may improve the model performance with proper knowledge about the DDI extraction task. Moreover, the current models find PPIs and DDIs that are given in a single sentence. Using an additional layer of hierarchy that represents document-to-sentence relations over the sentence-to-word update module, this work can be extended to extract relations between biomedical entities lying in different sentences.

5.3.6 Limitation

The model has achieved a significant performance boost. However, the trade-off is the computational time. Due to using two forward passes, the model requires more time to generate the results compared to the other models.

5.4 Conclusion

The experiments presented here for the protein-protein interaction (PPI) and drug-drug interaction (DDI) identification tasks result in two observations: firstly, the tree-structured neural

networks perform better than the sequential models for this task; and secondly, unlike some of the other prominent works, these tree-structured models do not require any additional features for the performance boost. As an example of additional features to improve the performance of the relation extraction model for the PPI and DDI tasks, Fei et al. [64] used multi-task learning and trained a named entity recognition model in combination with their graph-based relation extraction model. Instead, the constituency and dependency tree-transformers have the ability to use the syntactic and semantic information that they derive from the text by themselves to generate rich sentence representations for the relation extraction step. Moreover, the sentence-to-word update module, which uses the sentence representations generated by the tree-transformers, delivers task-specific contextual information, yielding enriched word embeddings that benefit the tree-transformers to produce even better sentence representations for PPI and DDI extraction.

Chapter 6

Personality Trait Identification

Identifying human personality traits is a multifaceted endeavour, characterized by the construction of various psychological models by experts. These models delineate personality traits as amalgams of distinct dichotomies. Our work addresses the challenge of personality trait identification through two distinct approaches, both aimed at extracting semantic nuances from textual data to discern the underlying dichotomies.

In the initial approach, we enhance vector representations of psycho-linguistic texts, leveraging a siamese architecture-based semantic similarity model to train statement encoders, thereby ensuring that the vector representations of given statements closely align with their respective baseline statements in the vector space. Our publication titled **"Interpretable Representation Learning for Personality Detection"** details this approach in Section 6.1.

In the second approach, we employ a multilabel classification strategy, which integrates insights from two of our publications: (i) **"Personality Trait Detection using an Hierarchy of Tree-transformers and Graph Attention Network"** and (ii) **"Detecting Personality Traits from Texts using an Hierarchy of Tree-Transformers and Graph Attention Network with Word Embedding Refinement"**. The former work, given in Section 6.2, utilizes tree-transformers to generate sentence representations for each sentence in a given statement. A graph attention network then integrates these sentence representations to create a more semantically enriched statement representation, enhancing the accuracy of the subsequent multi-label classification for personality trait prediction. The latter work, given in Section 6.3, extends the aforementioned approach by integrating phrase-level and inter-word information using constituency and dependency tree-transformers. A graph attention network is employed to generate context-enriched word representations which are subsequently fed into the hierarchy of tree-transformer and graph attention network layers to enhance statement representation. Our work has demonstrated state-of-the-art performance across three benchmark corpora spanning two distinct personality trait identification models.

6.1 Interpretable Representation Learning for Personality Detection

This section is based on the paper titled "Interpretable Representation Learning for Personality Detection" co-authored with Robert E. Mercer, Amirmohammad Kazemeini, and Erik Cambria that appeared in the *2021 International Conference on Data Mining Workshops (ICDMW)* [106].

Automatic personality detection has gained increasing interest recently. Several models have been introduced to perform this task. The weakness of these models is their inability to interpret their results. Even if the model shows excellent performance over test data, it can sometimes fail in real-life tasks since it may incorrectly interpret a statement. To investigate this issue, we evaluate two approaches. In the first approach we generate the sentence embeddings by training a siamese Bi-LSTM with max-pooling on the psychological statement pairs. The intent is to compute the semantic similarities between them. On the second approach we evaluate state-of-the-art pretrained language models to see whether their output representations can distinguish personality types. Both of these approaches outperform the previous state-of-the-art models for this task with less computational overhead. We conclude by discussing the implications of this work for both computational modelling and psychological science.

6.1.1 Introduction

AI has the potential to assist health experts in dealing with the increasing rate of mental health issues and disorders. This increasing trend has been the subject of recent investigations such as the recent trends in mental ill health and health-related behaviors in two cohorts of UK adolescents that show depressive symptoms and self-harm were higher in 2015 compared with 2005 [163]. How social media impacts mental health (including the mental health of adolescents and rising teen suicide rates) has also been studied [159]. This increasing rate of mental issues has accelerated due to the COVID-19 pandemic. According to a Kaiser Family Foundation poll, people have become more socially isolated and stressed. Nearly half of Americans report the coronavirus crisis is harming their mental health [3, 67].

According to a 2020 Harris Poll, between 46% and 51% of US adults were using social media more since the outbreak began [193]. Increased social media use means more digital footprints, and since people's personality and private traits can be identified based on them [111], this pandemic challenge can be turned into an advantage to provide more support for people based on their needs. A WHO survey showed that COVID-19 further burdened the already limited mental health services in many countries [233]. Since mental health service resources

are limited and mental health issues have increased, the increase in social media use provides an opportunity for AI researchers to utilize the produced digital footprints to help diagnose people's mental health issues.

Personality traits are defined as the set of relatively stable characteristics which describe our feelings and behaviour. These traits play important roles in individuals' futures and life outcomes [160, 185]. Among the various personality tests, the Big-Five, which is also called OCEAN, is known to be the most reliable test for assessing people's personality [97]. The OCEAN test describes personality in five measures: Openness, Conscientiousness, Extraversion, Agreeableness, and Neuroticism. Previous work has investigated the relationship between personality and mental disorders. Studies have shown that neuroticism plays a vital role in depressive and anxiety disorders [75].

Regarding the other traits, resilience demonstrates a strong inverse relationship with neuroticism and strong positive relationships with extraversion and conscientiousness and a small but statistically significant positive relationship with openness [38]. Hence, understanding a person's personality can provide a better insight for detecting mental illnesses.

In addition to psychological motivation, personality traits are also useful in recommender-systems [254, 141], product and service personalization [195, 226], job screenings [123], social network analysis [139], and sentiment analysis [37].

In this work, we address the following two questions: Does the embedding, which is used for current state-of-the-art model, capture psychological information? If not, how can it be improved? In order to answer these questions, we first introduce an approach for evaluating embeddings in personality detection. Following that, with metric learning in mind [103], we apply two different approaches using two siamese architectures for generating the embeddings from the psychological statements. The first approach produces sentence embeddings by means of computing semantic similarities between psychological statements representing different traits. In the second approach, different variants of another siamese sentence encoder, Sentence-BERT, for producing sentence embeddings for classifying psychological traits are investigated. Both of these approaches surpass the previous state-of-the-art models used in this task with the BFI statement data [95, 96, 23]. The second approach outperforms the previous state-of-the-art models with the Essays dataset [166] and the Kaggle personality dataset [99]. Extensive experiments with the Essays dataset and the BFI statements are performed and discussed. These experiments have focussed on these two datasets since the MBTI test (used in the Kaggle personality dataset) has been questioned for its comprehensiveness, dependability, and lack of independent categories [171], whereas the OCEAN personality test (the Essays dataset) is considered as more reliable. These approaches not only outperform the previous state-of-the-art model but also reduce the computational overhead.

6.1.2 Related Work

There are a variety of personality tests that are based on psychological discoveries [79]. The most accepted one in the field of psychology is the Big Five model, also called OCEAN [97]. This personality test is the one focussed on in this paper. OCEAN assesses five dimensions of personality (Openness to Experience, Conscientiousness, Agreeableness, Extraversion, and Neuroticism or when positively keyed, emotional stability). One other commonly used personality model, which is used in a comparison below, is Myers-Briggs, also known as MBTI [32]. MBTI categorizes personalities into 16 types; each one can be described as a combination of 4 binary categories (Extroversion/Introversion, Sensing/Intuition, Thinking/Feeling, Judging/Perceiving). Since the MBTI test has been questioned for its comprehensiveness, reliablity, and lack of independent categories, the OCEAN personality test is chosen as the main focus of this paper.

Given the limited mental health service resources, there is a strong need for an automated assistant tool. AI models have proven to be good candidates as they perform more accurately than humans in personality judgment [255]. Some models used psycholinguistic features to identify personality [135]. In the field of deep learning-based automatic personality detection, the hierarchical CNN model [136] has attracted a lot of attention. A full comparison between previous proposed models is given in [143] and perspectives are analyzed in [211]. Although the deep models are improving the accuracy in this field and their approaches have built the foundations of our current work, they suffer from some issues that prevent them from serving as well as they ought to. For example, the results might be based on the studied socio-cultural group. Lewis [117] has analyzed this diversity and has shown that the results can vary depending on the observed cohort. In addition, due to the delicate nature of mental health tasks, trust is an important criterion that these black-box models cannot satisfy without using a post-hoc explainability approach [183].

Current NLP models that understand human language are mostly proposed by large companies such as Facebook and Google, enabled by their high-spec infrastructure to create their high accuracy predictors [34, 56, 129]. Although they are not runnable on regular computers, their pre-trained versions can be used in personality detection with a small amount of fine-tuning to be adapted to this task [142, 225]. Considering that there is usually a trade-off between accuracy and simplicity, the task to obtain an optimal, yet simple model is non-trivial. Only a few papers, such as [105] (BB-SVM), have proposed high accuracy models in this field without sacrificing simplicity. BB-SVM also introduced a BERT-based personality model that can be used for longer sequences as well. However, even though this model is able to be run on ordinary computers, its interpretability, especially the justification for the choice of the pre-trained

model, has yet to be addressed.

First, as well as the existing trade-off between complexity and accuracy, a trade-off also exists between performance and transparency (i.e., explainability of the outcomes). The higher performing models tend to be more opaque [58]. As the model becomes more opaque, the need for explainability increases. To alleviate this problem, post-hoc explainability is used. This type of explainability is divided into model-agnostic approaches, which can be used for any model, and model-specific ones. A full comparison of explainable AI methods is given in [13].

Also, contemporary models learn from examples in specific datasets. This issue challenges the model when it faces new examples that are not the same as the previously observed ones since current models are not using experts' knowledge. So, even though the current models can do their best for their specific dataset, they cannot incorporate the socio-cultural diversity among groups of people, which results in the different ways they articulate their thoughts [117].

With the emergence of accurate AI models, theorists and researchers make normative claims based on the models' results [98]. Some of the previous experience has also shown how these models can be exploited for detrimental goals [78, 141]. Hence, by making the AI models more interpretable, more descriptive facts can be obtained based on their results. Ethical concerns can be slightly alleviated because of the insight which the model provides. [142] is one of the few works that address both improving personality detection accuracy using deep learning models and providing understandable insight using post-hoc explanablity approaches. This work is used as the baseline for the current paper.

6.1.3 Methodology

This section discusses the interpretable sentence representation generation approaches using the siamese architectures, the dataset we use for training the model, and the datasets used for evaluating the performance of the models. The sentence representation is generated by means of computing the semantic similarities between psychological statements. The reason behind choosing this approach is to preserve enriched semantics in the vector representations. Finally, the approach to interpret the output of the model is discussed along with the evaluation of the model. The interpretability of our approach is evaluated using the feature relevance and visual explanation methods of the post-hoc explainability category (see the taxonomy in Fig. 6 of [13]), by computing the cosine similarity between the input and baseline sentences and using PCA visualization, respectively.

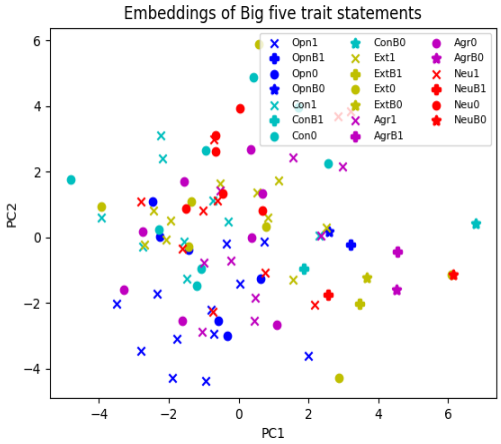

Figure 6.1: Visualization of the personality statements after applying PCA on the average of the output of layer 11 of Bert-base [142]. 1 and 0 mean "High" and "Low" rate of a specific trait, respectively, and "B" is for baseline sentences.

6.1.3.1 Datasets

We used the following publicly available personality datasets in our analyses:

6.1.3.1.1 Essays

This well-known stream-of-consciousness dataset consists of 2468 essays written by students and annotated with the binary labels of the Big Five personality traits which were found by a standardized self-report questionnaire [166].

6.1.3.1.2 Kaggle MBTI

This data was collected through the PersonalityCafe forum providing a diverse selection of people interacting in an informal online social setting. The dataset comprises 8675 records of each person's last 50 posts on the website along with their MBTI binary personality type [99].

6.1.3.2 Evaluating the Embeddings

In order to evaluate the pretrained BERT-base model for meaningful personality representations, we have used a simplified version of the Big Five Inventory (BFI) [95, 96, 23]. BFI is a self-report questionnaire that consists of 44 short phrases. Participants rate each of these statements based on their situation. Each statement focuses on assessing one of the five traits. We have simplified this version to make it easier for language models to extract meaningful representations from them. For example, the statement "I am someone who is talkative", which

Table 6.1: The baseline sentences for each trait of the Big Five personality test

Text	Trait	Label
I am extrovert	Ext	1
I am introvert	Ext	0
I am agreeable	Agr	1
I am disagreeable	Agr	0
I am neurotic	Neu	1
I am stable	Neu	0
I am an open person	Opn	1
I am not an open person	Opn	0
I am conscientious	Con	1
I am casual	Con	0

assesses the extraversion rate of a person, is converted to "I am talkative". In addition, to increase the dataset size, we have also added the adapted version of BFI [54, 76] to the original one. The final simplified statement set consists of 85 sentences, 44 of which belong to the original BFI statements and the rest are obtained from the adapted version. We have also used two baseline sentences for each trait. These sentences are listed in Table 6.1. We then use the pretrained version of BERT-base to extract the representations of the tokens. We have followed the best representation of Mehta et al. [142] which is averaging the output of the second to last layer to get the final representation of each statement. Next, we transform the embeddings using a PCA [1] with 2 principal components. The result of the PCA is illustrated in Fig. 6.1. The B-points are clustered in the upper half of the bottom right quadrant, whereas the 0- and 1-points are almost all in the left or upper quadrants. The representations of the baseline sentences are very close to each other and the distance between them and the corresponding trait statements are much larger. Hence, we can conclude that even when [142] gets high accuracy using these representations, it will not be generalizable since the extracted embeddings do not manifest the related personalities. Considering that this current state-of-the-art representation uses a rich corpus and state-of-the-art language models, we can infer that older ones probably also suffer from this issue. Furthermore, even if the baseline representations obtained from the previous methods maintain sufficient distance, their classification performance is worse compared to [142] which is also not acceptable. This motivates our investigating a model which cannot only improve the classification performance but also enhance explainability.

6.1.3.3 Interpretable Representation for Personality Detection

This paper investigates two different approaches for producing vector representations from psychological statements. The core idea behind both approaches is to use the extracted embed-

dings from the baseline sentences and BFI statements in order to evaluate the performance of the model. The output embedding can be explainable using this comparison.

Both of these approaches use siamese architectures using deep learning models. The first approach utilizes siamese Bi-LSTM with max-pooling over time of the output vectors. This model is trained on the simplified BFI statement pairs for computing the similarity between them. The second approach evaluates the Sentence-BERT variants [182]. The reason behind choosing the siamese models here is that we try to detect the personality traits not by applying direct classification approaches but rather by preserving the semantics of the statements where statements reflecting similar traits remain close to each other in the embedding space. This objective is achieved by leveraging psychological datasets (the BFI statements and the baseline sentences).

6.1.3.3.1 Bi-LSTM with Max-pooling

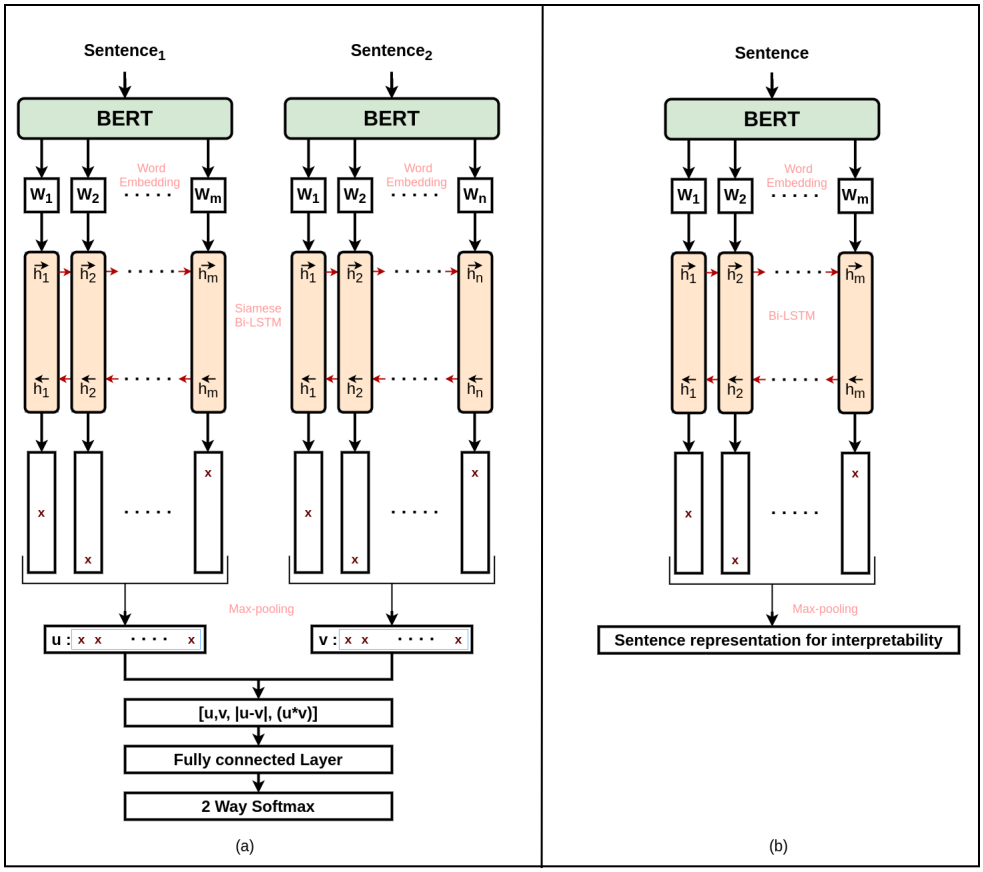

Figure 6.2: Architecture of the model with siamese Bi-LSTM and max-pooling for the interpretable tool for personality detection. (a) The training of the model, (b) After training, the Bi-LSTM followed by the max-pooling layer act as the sentence encoder.

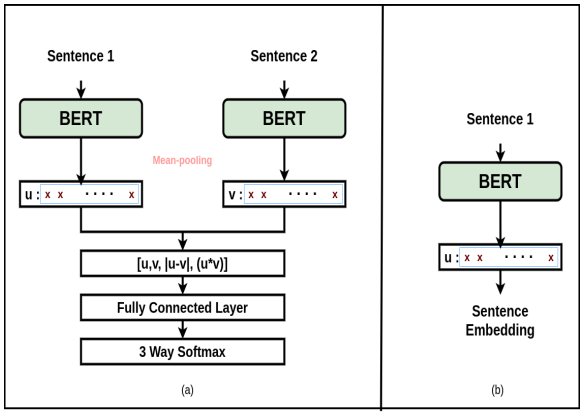

Figure 6.3: Architecture of Sentence-BERT. (a) Training of the model on the natural language inference datasets. (b) Sentence encoder.

To extract the feature vectors of both the BFI statements and the baseline sentences, we have used the siamese architecture of Bi-LSTM over the BERT word embeddings from layer 11 of BERT-base. The architecture is inspired by the InferSent model [53]. The basic idea of this model is to generate a sentence embedding by means of computing the semantic similarity between two sentences. This semantics attempts to preserve the personality trait from the BFI statement.

For the word embeddings we have chosen the output of layer 11 of the pre-trained BERT-base. For any given sentence pair, word embeddings are fed to two identical Bi-LSTMs. These Bi-LSTMs share the same parameters and weights. For a sequence of N words, Bi-LSTM produces a set of N vectors. The final hidden state representation for each time step is generated by concatenating the hidden representation of the forward ($\overrightarrow{h_i}$) and backward LSTMs ($\overleftarrow{h_i}$) [196]. For each time step, max-pooling is applied over these concatenated hidden representations ($[\overrightarrow{h_i}, \overleftarrow{h_i}]$) to generate an intermediate sentence representation. In the next step, three operations, concatenation, point-wise difference and point-wise multiplication, are performed on the representations obtained for both of the sentences from the sentence pair. Finally, the outcome of these three matching operations are concatenated and fed to a feed-forward neural network for classification like [53]. Suppose, u and v are the intermediate representations for the sentences after max-pooling. Then $[u, v, |u - v|, (u * v)]$ would be the final feature representation to be fed to the following classifier. The classifier outputs either 0 or 1 where 1 indicates the sentences offer semantically similar traits and 0 otherwise. Fig. 6.2 portrays the overall architecture of the model. After the training is done, the Bi-LSTM together with the max-pooling layer acts as the encoder for generating the sentence representation. This representation is a 768 dimensional vector.

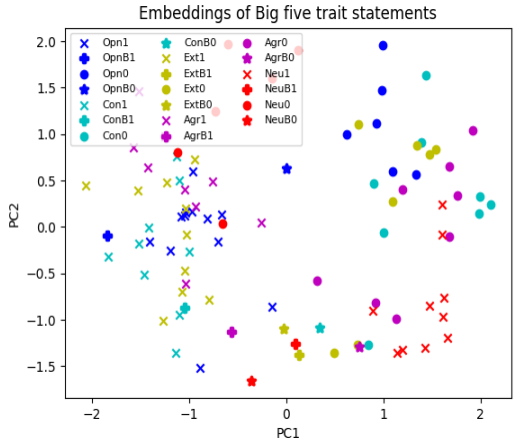

Figure 6.4: Visualization of the personality statements after applying PCA on the feature vectors of Bi-LSTM and max-pooling. 1 and 0 mean "High" and "Low" rate of a specific trait, respectively, and "B" is for baseline sentences.

6.1.3.3.2 Sentence-BERT

Sentence-BERT [182] is a refinement of the pretrained BERT using siamese and triplet structures. It can derive sentence representations preserving the semantics of the sentences. Unlike BERT, which outputs rich token embeddings and [CLS] with poor semantics for the sentence, Sentence-BERT produces semantically richer sentence embeddings. It is trained on the sentence pairs from the SNLI dataset [31] and multi-genre NLI dataset [240]. It has been shown that sentence embedding models trained on natural language inference datasets have better semantic preserving abilities [53]. For this reason, Sentence-BERT outputs semantically richer sentence embeddings.

Sentence-BERT incorporates a mean-pooling operation over the output of each BERT embedding to generate two sentence embeddings for the sentence pair. Then two matching operations, concatenation and point-wise difference, are performed on them. Finally, this feature is fed to the softmax classifier. After the fine tuning is complete, the fine-tuned BERT with the mean-pooling act as the sentence encoder. Using this pretrained Sentence-BERT is then a straight-forward approach. After being given a sentence, it directly outputs the corresponding 768 dimensional vector sentence embedding. The architecture of Sentence-BERT is shown in Fig. 6.3. We have conducted experiments on the Essays, the BFI statements, and the Kaggle datasets using different variants of Sentence-BERT [181]. In all cases the overall architecture remains the same, only the BERT encoder is varied. Some prominent variants are RoBERTa [129] and MPNet [210].

6.1.4 Experiments

To analyze the effectiveness of our siamese Bi-LSTM model, for each personality trait t, we create all possible corresponding BFI statement pairs together with the appropriate label, $(s_i, s_j, l_{i,j})$, where $l_{i,j}$ is 1 if the statements s_i and s_j have the same label and 0 if s_i and s_j have different labels. Then, we feed the statement pairs as inputs to the model and use $l_{i,j}$ as the label which the model tries to predict. Applying this approach over the BFI statements, the data set has 681 sentence pairs. Among these, 600 samples are used for training and the remaining 81 are used for validation. This small dataset was sufficient for training the siamese LSTM model with some good training and validation accuracies. While testing this model on the BFI statements, it achieved a better result compared to the previous models [142]. This comparison is performed using the *PredLabel* and *SimScore* metrics. In addition, the finetuned embedding are also assessed by replacing the embedding part of the model in [142] for classifying the Kaggle and Essays datasets. However, the model trained on this data did not achieve state of the art accuracies as the training data was comparably small.

We have trained the siamese Bi-LSTM model for only 25 epochs where the best result was found at the 21st epoch. While training, the batch size was set to 10 with 10% dropout. Standard gradient descent was used for optimization with a learning rate $1e^{-5}$. The forward and backward LSTMs' hidden representations are 384 dimensional vectors.

After the training phase, we use the feature vectors extracted from the Bi-LSTM for evaluation as we did in Section 6.1.3.2 for the BFI statements. After extracting the feature vectors of both the BFI statements and the baseline sentences, for each statement that belongs to trait t we assign a similarity score and prediction label based on the closeness to the corresponding baseline sentences as following:

$$\forall s_i \in S_t : SimScore(s_i) = \\ (-1)^{l_i - 1} C(s_i, b_{t,1}) + (-1)^{l_i} C(s_i, b_{t,0})$$

and

$$PredLabel(s_i) = \begin{cases} 1, & \text{if } C(s_i, b_{t,1}) > C(s_i, b_{t,0}) \\ 0, & \text{otherwise} \end{cases}$$

where l_i is the label of s_i, C is cosine similarity, and $b_{t,0}$, $b_{t,1}$ are the baseline feature vectors of trait t. To report the result of a specific model, we use accuracy for the *PredLabel*s and the average of the *SimScore*s. For the Sentence-BERT models, the BFI statements and baseline statements are fed to the pretrained encoders and then the accuracy of the *PredLabel*s and the average of the *SimScore*s are computed. While testing, we aggregated both the simplified and non-simplified versions of the BFI statements to generate a more generalized model. The

Table 6.2: Comparison of accuracies of *PredLabel*s of different representations.

Model	O	C	E	A	N	Average
BERT (average) [142]	61.11	52.94	41.18	64.71	56.25	55.24
BERT (CLS)	33.33	58.82	41.18	47.06	62.5	48.58
Bi-LSTM with max-pooling	94.44	100.00	32.35	100.00	53.13	75.98
average_word_embeddings_glove.6B.300d	33.33	58.82	70.59	76.47	43.75	56.59
average_word_embeddings_glove.840B.300d	33.33	64.71	88.24	70.59	62.50	63.87
average_word_embeddings_komninos	33.33	70.59	76.47	70.59	75.00	65.20
average_word_embeddings_levy_dependency	33.33	41.18	47.06	64.71	62.50	49.76
nli-bert-base	66.67	76.47	70.59	88.24	100.00	80.39
nli-bert-base-cls-pooling	77.78	76.47	70.59	88.24	93.75	81.36
nli-bert-base-max-pooling	77.78	88.24	70.59	88.24	93.75	83.72
nli-bert-large	94.44	94.12	100.00	88.24	93.75	94.11
nli-bert-large-cls-pooling	88.89	88.24	100.00	88.24	100.00	93.07
nli-bert-large-max-pooling	88.89	82.35	100.00	88.24	100.00	91.90
nli-distilbert-base	72.22	88.24	17.65	88.24	93.75	72.02
nli-distilbert-base-max-pooling	77.78	82.35	11.77	88.24	87.50	69.53
nli-distilroberta-base-v2	72.22	94.12	70.59	88.24	100.00	85.03
nli-mpnet-base-v2	100.00	88.24	94.12	94.12	93.75	94.04
nli-roberta-base	94.44	82.35	100.00	88.24	93.75	91.76
nli-roberta-base-v2	83.33	94.12	100.00	88.24	100.00	93.14
nli-roberta-large	100.00	100.00	100.00	88.24	100.00	**97.65**
paraphrase-distilroberta-base-v1	33.33	70.59	47.06	70.59	87.50	61.81
paraphrase-xlm-r-multilingual-v1	83.33	70.59	47.06	76.47	93.75	74.24
stsb-bert-base	72.22	76.47	76.47	76.47	87.50	77.83
stsb-bert-large	88.89	88.24	100.00	82.35	68.75	85.65
stsb-distilbert-base	72.22	88.24	29.41	82.35	93.75	73.19
stsb-distilroberta-base-v2	72.22	82.35	70.59	82.35	100.00	81.50
stsb-mpnet-base-v2	94.44	94.12	94.12	100.00	93.75	95.29
stsb-roberta-base	100.00	70.59	76.47	82.35	100.00	85.88
stsb-roberta-base-v2	88.89	70.59	88.24	88.24	100.00	87.19
stsb-roberta-large	100.00	94.12	76.47	88.24	100.00	91.77

embeddings of the BFI and the baseline statements are extracted from the encoder portion of the siamese Bi-LSTM as previously described and finally, *PredLabel*s and *SimScore*s are measured.

In the case of experimenting with the Essays dataset, no further training is performed. The statements are fed to the models (both the Bi-LSTM with max-pooling and the Sentence-BERTs). Then they are tested against the baseline statements to compute the performance metrics. The Kaggle dataset is tested with the Sentence-BERTs only.

6.1.5 Results

The accuracies of the *PredLabels* are shown in Table 6.2, and the *SimScore*s for the BFI statements, in Table 6.3. For three traits, Bi-LSTM with max-pooling outperforms the CLS and

Table 6.3: Comparison of *SimScore*s of different representations.

Model	O	C	E	A	N	Average
BERT (average) [142]	0.011	0.007	-0.003	0.026	0.002	0.009
BERT (CLS)	0.001	0.001	-0.011	0.012	0.009	0.002
Bi-LSTM with max-pooling	0.082	0.064	-0.01565	0.079	0.008	0.044
average_word_embeddings_glove.6B.300d	0.000	0.039	0.038	0.066	0.011	0.031
average_word_embeddings_glove.840B.300d	0.000	0.036	0.040	0.082	0.077	0.047
average_word_embeddings_komninos	0.000	0.036	0.031	0.039	0.077	0.036
average_word_embeddings_levy_dependency	0.000	-0.007	0.002	0.020	0.075	0.018
nli-bert-base	0.124	0.148	0.073	0.253	0.321	0.184
nli-bert-base-cls-pooling	0.145	0.134	0.063	0.277	0.330	0.190
nli-bert-base-max-pooling	0.116	0.141	0.035	0.187	0.224	0.141
nli-bert-large	0.231	0.211	0.160	0.270	0.211	0.217
nli-bert-large-cls-pooling	0.224	0.166	0.159	0.281	0.304	0.227
nli-bert-large-max-pooling	0.163	0.169	0.246	0.283	0.264	0.225
nli-distilbert-base	0.068	0.149	-0.088	0.194	0.224	0.109
nli-distilbert-base-max-pooling	0.088	0.147	-0.082	0.162	0.166	0.096
nli-distilroberta-base-v2	0.037	0.119	0.046	0.180	0.181	0.112
nli-mpnet-base-v2	0.148	0.086	0.209	0.253	0.223	0.184
nli-roberta-base	0.194	0.158	0.142	0.228	0.356	0.215
nli-roberta-base-v2	0.160	0.117	0.138	0.206	0.226	0.169
nli-roberta-large	0.248	0.278	0.245	0.274	0.415	**0.292**
paraphrase-distilroberta-base-v1	0.020	0.025	0.002	0.060	0.080	0.037
paraphrase-xlm-r-multilingual-v1	0.032	0.030	-0.004	0.074	0.117	0.050
stsb-bert-base	0.158	0.129	0.150	0.200	0.212	0.170
stsb-bert-large	0.251	0.174	0.145	0.261	0.140	0.194
stsb-distilbert-base	0.119	0.163	-0.041	0.221	0.272	0.147
stsb-distilroberta-base-v2	0.045	0.131	0.039	0.196	0.227	0.128
stsb-mpnet-base-v2	0.174	0.081	0.206	0.191	0.179	0.166
stsb-roberta-base	0.259	0.095	0.152	0.305	0.352	0.233
stsb-roberta-base-v2	0.107	0.097	0.122	0.190	0.243	0.152
stsb-roberta-large	0.218	0.262	0.077	0.226	0.315	0.219

average methods of BERT which were used in Mehta et al. [142]'s state-of-the-art model for this task and outperforms on the average result as well. For each of the personality traits, the 0- and 1-statements form distinguishable and well-separated clusters except for the Neuroticism and Extroversion baseline sentences, which are so close to each other. The PCA result is illustrated in Fig. 6.4. The evaluation also tries to identify whether the model is able to assign the correct binary trait label to the statements. For Openness, Conscientiousness, and Agreeableness, as it is shown in Fig. 6.4, the model can almost completely understand which statement belongs to which baseline trait. Regarding Neuroticism, although the *SimScore* is better than both the CLS and the average methods, the classification metric was not satisfactory. Extraversion also seems to be the most difficult trait to be identified by baseline sentences. Although the statements are separated, the embeddings of "I am extrovert" and "I am introvert" are still

Table 6.4: Accuracy of Bi-LSTM with max-pooling and Sentence BERT models on Essays and Kaggle datasets.

MODEL	Essays						Kaggle MBTI				
	O	C	E	A	N	Average	I/E	N/S	T/F	P/J	Average
Majority Baseline	51.5	50.8	51.7	53.1	50.0	51.4	77.0	85.3	54.1	60.4	69.2
BERT-base [142]	64.6	59.2	60.0	58.8	60.5	60.6	78.3	86.4	74.4	64.4	75.9
BERT-large [142]	63.4	58.9	59.2	58.3	58.9	59.7	78.8	86.3	76.1	67.2	77.1
Bi-LSTM max-pooling_combined	61.7	54.6	55.0	56.7	55.9	56.8	-	-	-	-	-
average_word_embeddings_glove.6B.300d	63.2	58.5	56.3	57.2	58.5	58.7	77.2	86.5	76.9	66.2	76.7
average_word_embeddings_glove.840B.300d	63.0	58.0	57.2	57.5	57.7	58.7	78.6	87.1	**79.6**	68.6	78.5
average_word_embeddings_komninos	62.5	57.9	55.3	56.6	58.5	58.1	77.0	86.2	74.3	63.0	75.1
average_word_embeddings_levy_dependency	61.4	55.9	54.0	53.3	56.7	56.3	77.0	86.2	70.2	60.5	73.5
nli-bert-base	64.0	**60.0**	58.7	58.2	60.4	60.2	77.6	86.4	70.8	62.5	74.3
nli-bert-base-cls-pooling	63.8	59.7	57.7	59.1	60.1	60.1	77.6	86.3	71.1	62.2	74.3
nli-bert-base-max-pooling	63.0	58.0	56.7	57.4	58.4	58.7	77.5	86.2	69.7	61.8	73.8
nli-bert-large	63.5	59.8	57.1	58.7	60.8	60.0	77.6	86.3	71.2	62.2	74.3
nli-bert-large-cls-pooling	63.6	59.2	57.9	58.7	60.1	59.9	77.5	86.3	71.3	62.7	74.4
nli-bert-large-max-pooling	63.0	58.1	58.3	58.5	59.1	59.4	77.5	86.2	70.8	61.9	74.1
nli-distilbert-base	62.5	58.8	58.5	57.8	59.4	59.4	77.6	86.2	71.4	62.3	74.4
nli-distilbert-base-max-pooling	62.4	57.0	57.5	57.5	60.2	58.9	77.5	86.2	68.8	61.7	73.6
nli-distilroberta-base-v2	63.2	58.5	59.5	58.7	**61.5**	60.3	81.0	87.3	77.9	**71.5**	**79.4**
nli-mpnet-base-v2	64.2	58.8	59.7	59.1	60.6	60.5	81.0	87.2	78.1	69.3	78.9
nli-roberta-base	62.0	59.1	58.9	59.2	59.0	59.6	77.7	86.3	72.0	62.4	74.6
nli-roberta-large	63.9	59.5	**60.2**	**59.5**	61.3	**60.9**	80.7	87.2	77.7	70.9	79.1
nli-roberta-base-v2	62.8	59.7	58.9	59.3	60.8	60.3	77.9	86.5	72.0	63.1	74.9
paraphrase-distilroberta-base-v1	**65.0**	57.8	59.3	59.0	59.7	60.2	80.1	87.1	76.2	70.7	78.5
paraphrase-xlm-r-multilingual-v1	63.6	58.1	58.8	57.3	59.8	59.5	79.1	86.6	74.2	67.8	77.0
stsb-bert-base	64.0	59.1	57.7	58.1	60.6	59.9	78.1	86.5	72.4	63.4	75.1
stsb-bert-large	62.4	56.9	58.0	58.1	61.4	59.4	77.5	86.5	71.3	62.4	74.4
stsb-distilbert-base	62.8	58.0	58.0	57.1	59.3	59.1	78.5	86.5	73.1	64.6	75.7
stsb-distilroberta-base-v2	63.8	58.9	58.5	58.9	59.8	60.0	**81.1**	87.2	77.3	71.0	79.2
stsb-mpnet-base-v2	64.2	58.6	58.7	59.0	61.1	60.3	**81.1**	**87.5**	78.0	69.1	78.9
stsb-roberta-base	63.4	58.2	57.4	57.8	59.5	59.3	80.3	86.8	76.1	65.8	77.2
stsb-roberta-base-v2	63.4	58.7	59.7	58.9	60.6	60.3	81.0	87.3	77.5	70.3	79.0
stsb-roberta-large	62.7	58.4	57.6	58.0	59.7	59.3	80.1	86.6	74.2	65.4	76.6

Table 6.5: The Pearson correlation between the Predlabel accuracy and the Essays accuracy for all Sentence-BERT embeddings. *p <.05. **p <.001, two-tailed.

O	C	E	A	N	Ave.
0.086	0.488*	0.208	0.662**	0.533**	0.700**

too close, resulting in the poor result. We believe this issue happens because of the dataset which is used for training BERT.

Overall, since we have not used the baseline sentences in any phase of the training process, and they are used only in the evaluation, we believe that Bi-LSTM with max-pooling has used the general language model knowledge enriched with knowledge from the psychological statements to distinguish between traits. Average results have shown that this model is successful in learning the personality trait-specific representations while retaining its knowledge from the pre-trained BERT.

Even though the Bi-LSTM with max pooling outperforms the previous state-of-the-art

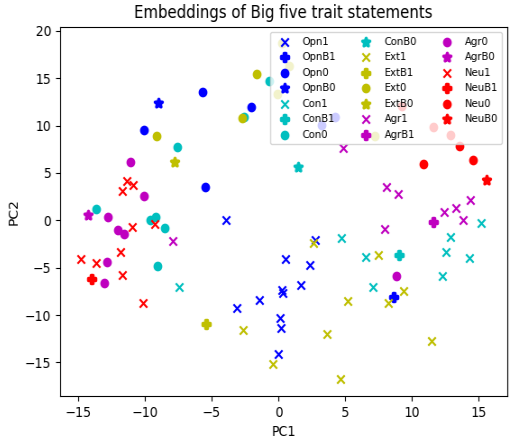

Figure 6.5: Visualization of the personality statements after applying PCA on nli-roberta-large version of Sentence BERT. 1 and 0 mean "High" and "Low" rate of a specific trait, respectively, and "B" is for baseline sentences.

when compared by performance metrics as well as richer personality trait-specific representation generation, the Sentence-BERT based model outperforms this one. We have experimented with different variants of Sentence-BERT. Among them, the most prominent results are found when RoBERTa-large or MPNet are used as the encoder in the Sentence-BERT architecture. In terms of accuracy of the *PredLabel*s and *SimScore*s, overall, RoBERTa-large performs the best. It achieves an accuracy for *PredLabel* of 97.65% which is almost double the previous state-of-the-art model's accuracy [142]. Apart from Agreeableness, its *PredLabel* accuracy is 100%, whereas for Agreeableness, it's 88.24%. MPNet achieves 100% *PredLabel* accuracy for Agreeableness. On average MPNet achieves 95.29% *PredLabel* accuracy. In terms of *SimScore*s, RoBERTa-large performs the best in all cases apart from Agreeableness. Still, its average value, 0.292, is more than three times that of [142]'s result. For Agreeableness, the encoder with MPNet performs the best for *SimScore*, 0.305, and on average it achieves 0.233. Fig. 6.5 portrays in a 2D projection the representations generated by the RoBERTa-large version of Sentence-BERT, showing that the closeness of each statement to any particular trait is very clear. For each of the personality traits, the 0-, 1- and B-statements form distinguishable and well-separated clusters (with a couple of exceptions) as demonstrated. One issue of note, two metrics, *PredLabel* and *SimScore*, are used to measure the performance of the model. PCA has been used only to provide a visualization of the embeddings to show how close the representations of the similar trait samples are. We have also used other visualization techniques like t-SNE, UMAP, and LDA. As all the visualization results are very similar, we have reported only the PCA visualization.

To evaluate the generalizability of the model, we tested these models on the Essays and

the Kaggle personality datasets. This time the Bi-LSTM with max-pooling performs worse than [142]'s work. The overall accuracy is almost 2% lower for the Essay dataset. But this is justifiable as this Siamese model was trained on very short sentences from the BFI statements, whereas the Essays dataset comes with long paragraphs. Additionally, LSTM based models face shortcomings while working with very long sequences. But the Sentence-BERT models, without any kind of additional operations, outperform the BERT-based averaging technique [142]. This time, RoBERTa-large achieves 60.9% accuracy which is an almost 1 percentage point boost compared to the previous works. In the case of the Kaggle personality dataset, RoBERTa-large gains almost 2 percentage points more accuracy (79.1%). However, Distil-RoBERTa performs the best for this dataset and achieves 79.4% accuracy. In both cases, MP-Net shows prominent results with accuracies 60.3% and 78.9%, respectively.

We also computed the Pearson correlation of the accuracy of PredLabel and Essays to see if the PredLabel accuracy gives any insight into how an encoder works for real world datasets. As demonstrated in Table 6.5, although the experimented encoders are not specifically designed for long sequence datasets such as Essays, for most traits, especially the average of the traits, there is a significant positive correlation between these two accuracies. Hence, we can conclude that using PredLabel is a good approach for picking the best encoder for real-life datasets.

One notable significance of these models is that none of them have been enhanced with any kind of additional psychological features, unlike [142]. While training, the models are simply trained with sentence pairs. Thus it reduces the computational overhead as well. And as RoBERTa-large was initially trained over larger sequences and then fine-tuned again over natural language inference data, Sentence-BERT with RoBERTa-large earns the capability to produce sentence embeddings preserving richer semantics than the others. Furthermore, as the Sentence-BERT models are trained on a very large corpus of real life inference data compared to the siamese LSTM model, which is trained on the small BFI statement pairs dataset, they have achieved the ability to provide better representations of the statements.

6.1.6 Conclusion

In this paper, we analyze the weakness of the state-of-the-art personality detection model. In addition, with computationally less overhead our model delivers sentence embeddings for psychological statements with rich semantics. Our results show that our enriched representations distinguish the personality traits better than the CLS and average methods which are common in the field. Furthermore, we have used the enriched representations in addition to Sentence-BERT models to classify traits based on their closeness to the baseline psychological statements so the result can be regarded as interpretable. Our experiments improved the Kaggle

state-of-the-art accuracy by 2.3 percentage points and Essays by 0.3 percentage points. This work restricts the statements at the sentence level. In future it can be extended to the paragraph level using hierarchical models like SMITH [246] so that better representations from the paragraphs can also be captured. Besides, BFI statements can be used within the prediction model to identify the closeness of each of the samples in the dataset with each of the BFI statements. We believe this method will help psychologists to get better insights into the prediction.

6.2 Personality Trait Detection using an Hierarchy of Tree-Transformers and Graph Attention Network

This section is based on the paper titled "Personality Trait Detection using an Hierarchy of Tree-transformers and Graph Attention Network" co-authored with Robert E. Mercer, and Souvik Kundu that appeared in the *2023 Canadian Conference on Artificial Intelligence* [207].

Automatic personality trait detection from a person's writings is helpful for professionals to assess the mental health of an individual, as well as helping individuals to determine their strengths and weaknesses for making choices such as personal improvement, workplace compatibility, and life-style decision-making. Psychologists have identified a set of personality traits that may be present in an individual's personality. This work classifies the writings of an individual into a subset of these traits. The classifier model comprises an hierarchical structure of tree-transformers and a graph attention network (GAT). The tree-transformers encode the sentences and the following GAT layer encodes the complete text of an individual's writing. Our model has shown a large performance boost over two benchmark corpora compared to previous works.

6.2.1 Introduction

Artificial intelligence (AI) has become a valuable tool for aiding psychiatrists and healthcare professionals in addressing the growing incidence of mental health related issues and disorders [106]. This upward trajectory has garnered recent attention, with studies like "Changes in Mental Ill Health and Health-Related Behaviors in Two Cohorts of UK Adolescents" revealing that rates of depression symptoms as well as self-harm tendencies have risen to multiple times in 2015 compared to 2005 [163]. In addition, research has examined the effect of social media on mental health, including its impact on adolescents' mental health and the increasing prevalence of teen suicide [159].

The COVID-19 pandemic has exacerbated the rising incidence of mental health concerns. A Kaiser Family Foundation survey has reported that individuals have become more distressed

and disconnected from their social life, with nearly 50% residents of America reporting that the pandemic has negatively impacted their mental wellbeing [3, 67].

A 2020 Harris Poll [193] shows social media usage has increased among US adults, about 50% reporting higher usage during the pandemic. This trend was particularly noticeable among younger age groups, with 60% of those aged 18 to 34, 64% of those aged 35 to 49, and 34% of those aged 65 and older reporting increased social media usage [108].

Personality traits refer to a collection of enduring qualities, rooted in psychological research [79], that define an individual's emotions and actions in a relatively consistent manner. The Big-Five personality traits (also called OCEAN) is the best accepted and most commonly used model of personality [97]. OCEAN describes personality with these five measures: Openness to Experience, Conscientiousness, Extraversion, Agreeableness, and Neuroticism (or positively keyed as emotional stability) [106]. Another frequently used personality model, the Myers-Briggs Type Indicator (MBTI) [32], categorizes 16 types of personalities characterized by a combination of four binary categories: Extroversion or Introversion, Sensing or Intuition, Thinking or Feeling, and Judging or Perceiving. These traits play important roles in an individual's future and life outcomes [160, 185].

The upsurge in social media activity during the pandemic has resulted in more digital footprints being left behind. These footprints can reveal an individual's personality and emotional traits, as has been demonstrated by Kosinski et al. [111]. This presents an opportunity to leverage these data to provide tailored support to individuals based on their unique needs, thus transforming the pandemic challenge into a potential advantage in terms of mental health care.

Many countries have faced additional burden on their mental health services due to the COVID-19 pandemic, as highlighted by a survey conducted by the World Health Organization (WHO) [233]. Given the scarcity of mental health service resources and the surge in mental health issues, the rise in social media usage presents an window of opportunity for AI researchers to leverage the resulting digital footprints to aid in diagnosing individuals' mental health concerns.

Prior research has explored the connection between personality traits and mental health disorders. Several studies have evidenced that neuroticism is a crucial factor in the development of depression and anxiety disorders [107, 75]. In addition, studies have found that resilience is inversely correlated with neuroticism and positively associated with conscientiousness and extraversion. Moreover, The positive correlation between openness and resilience is modest, but significant statistically [38]. Thus, automatic comprehension of an individual's personality can have a significant impact on the treatment process for mental health concerns. This has the potential to improve treatment outcomes and alleviate the burden on mental health services.

In this study, we have developed two deep-learning models that integrate tree-transformers

[6] and graph attention networks [224]. The aim is to generate more nuanced vector representations of statements, preserving their underlying semantics, to facilitate the identification of personality traits through subsequent classification. To evaluate the efficacy of our approach, we have conducted experiments on various benchmark corpora, where statements are labeled with one or more personality traits. Our experimental results demonstrate that our model surpasses the performance of many state-of-the-art methods.

6.2.2 Related Works

Due to limited mental health resources compared to the demand in current time, automated assistant tools can be a great support to help diagnose mental health issues and AI models have the potentiality to offer a great help. AI models have shown promise as automated assistants for such services due to their superior performance in personality judgment compared to humans [255]. Numerous studies have effectively utilized machine learning techniques to identify personality traits in social media content [219, 40]. Detecting personality traits can be achieved through various features, including demographic data and text data (e.g., self-interpretation and content from social media). One early example is Argamon et al.'s [12] model, which has used support vector machines (SVMs) and statistical features extracted from functional lexicons to identify personality traits. Farnadi et al. [63] built on the work of Argamon et al. [12] and used SVM to detect personality traits based on features such as network size, density, and frequency of updating status. Zhusupova et al. [275] utilized social media activity and demographic data to detect the personality traits of Twitter users from Portugal.

In recent years, several notable works have employed various deep learning models for the task of identifying personality traits. Kalghatgi et al. [101] used neural networks, specifically multilayer perceptrons (MLP), along with hand-crafted features to detect personality traits. Meanwhile, Su et al. [212] employed recurrent neural networks (RNN) and hidden Markov models (HMM) to identify personality traits from Chinese Language Inquiry and Word Count (LIWC) annotations extracted from dialogues. Tandera et al. [215] and Sun et al. [213] employed long-short-term-memory (LSTM) and convolutional neural network (CNN) to detect personality traits directly from text data collected from facebook posts. Meanwhile, Liu et al. [124] developed a hierarchical structure based on Bidirectional recurrent neural network to learn representations of words and sentences that can predict personality traits from multilingual (English, Spanish, and Italian) statements. After experimenting on 275 LinkedIn profiles, Van de Ven et al. [221] have demonstrated with evidence that extroversion can be accurately inferred from self-description in the user profiles. Lynn et al. [133] used message level attention over facebook posts to analyze users' personality traits. Majumder et al. [136]

have utilized psycholinguistic features [135] and deep learning models, such as hierarchical CNN, for automatic personality detection. Gjurković et al. [74] utilized Sentence-BERT [182] over their self-created corpus. Kazameini et al. [105] has applied an ensemble of SVMs over BERT embeddings and achieved better performance compared to other models on the Essays corpus [166] for Big Five trait classification. Mehta et al. [142] has experimented with various combinations of BERT-based models and psycholinguistic features and analyzed each feature's impact on the trait prediction. They have also achieved state-of-the-art results on different corpora. A comprehensive analysis of previous models is presented in [143], while a review of perspectives is discussed in [211].

While these models have improved the accuracy over time, they face several limitations that hinder their effectiveness in practice. The major reason is that textual representations are complex in nature and that the word level dependencies between long distant words plus the constituency representations mean a lot while generating the semantics. The use of only sequential models cannot capture this information appropriately. Again, though the pre-trained language model-based approaches have achieved state-of-the-art results over the benchmark personality trait classification corpora, they are limited to handle 512 words from the statements as these language models can take a maximum of 512 input tokens. This is definitely a hindrance for real-life applications of these models as automated assistant tools. Considering these two issues, we have investigated a model which utilizes tree-transformers to utilize word-level dependency and phrasal information followed by a graph attention network (GAT) to combine the sentence representations when generating the full statement representation. By using the tree-transformers and the GAT, this approach has the ability to preserve the syntactical structure of the sentences, and at the same time is not restricted to word limits of each sentence and the text as a whole, like Kazameini et al. [105] and Mehta et al. [142].

6.2.3 Methodology

The personality trait classification model is built upon an hierarchical structure consisting of a sentence encoder and then a full statement encoder. The sentence encoder unit works over the words and generates a vector for each sentence in the text. Then the statement encoder unit generates a vector representation of the complete text from the individual sentences. To utilize the syntactical information that is present in the textual representation, as the sentence encoder we have experimented with two types of tree-transformers: the constituency and dependency tree-transformers. For combining these sentence representations to obtain the statement representation, we have utilized a graph attention network (GAT). This section first talks about the individual building blocks, providing a better understanding of the concepts, and then it

describes the proposed model as a whole.

6.2.3.1 Sentence Encoding Module

To analyze an individual's personality traits from texts, we need to consider the syntactical structures of the sentences as the sentence representations play a crucial role while generating the whole statement embedding. To serve this purpose, we have investigated two types of tree-structured transformer models in this work as Tai et al. [214] has showed that tree structured representations are a better fit while working with text data compared to sequential representations. Sequential models are not capable enough to consider correlations between long distant words and phrasal representations present in the sentence. Attention [18, 223] goes a long way to solving the problem which occurs due to the long distance between the considered words. However, they are not capable of competing with tree-structured models [6, 5] which also take into account the relationships between words and the phrases that words make up.

There are two types of tree-based representations used to convey information about a sentence: constituency and dependency trees. These representations capture different aspects of the sentence's syntax, with constituency trees representing the structure of phrases and dependency trees illustrating the relationships between individual words located at different positions in the sentence. In a study by Ahmed et al. [6], two tree-transformer models have been proposed to make use of this syntactic structure information: a constituency tree-transformer and a dependency tree-transformer. The aim of these models is to carefully examine each sub-tree inside a constituency or dependency tree structure and recursively compute the root of each sub-tree to generate a sentence vector representation at the root of the tree through attentive processing over branches.

In a dependency tree, each node corresponds to a word in the sentence. When traversing a sub-tree in this type of tree, the dependency tree-transformer takes into account the representations of both the parent and child nodes. On the other hand, a constituency tree only has words at the leaf nodes, while the vectors for non-terminal nodes are computed only after the full traversal of the sub-tree is completed.

Ahmed et al. [6] have enriched the dependency and constituency tree representations of a sentence by using self-attention over the branches, which involves computing *query* ($\mathcal{Q}$), *key* ($\mathcal{K}$), and *value* ($\mathcal{V}$) matrices. The matrices are computed in the following way [223]:

$$\mathcal{K} = \omega_k \mathcal{M}_k \quad \text{s.t.} \quad \omega_k \in \mathbb{R}^{d \times d} \tag{6.1}$$

$$\mathcal{V} = \omega_v \mathcal{M}_v \quad \text{s.t.} \quad \omega_v \in \mathbb{R}^{d \times d} \tag{6.2}$$

$$\mathcal{Q} = \omega_q \mathcal{M}_q \quad \text{s.t.} \quad \omega_q \in \mathbb{R}^{d \times d} \tag{6.3}$$

To create the matrix $\mathcal{M}$ in a dependency tree, the word vectors of all child nodes corresponding to each parent node are concatenated. For a constituency tree, $\mathcal{M}$ is formed by concatenating the word vectors within a constituent. The tree-transformer models use the $\mathcal{Q}$, $\mathcal{K}$, and $\mathcal{V}$ matrices to compute the self-attention matrix in the following manner:

$$\alpha = \text{softmax}(\frac{\mathcal{Q}\,\mathcal{K}^T}{\sqrt{d_k}})\mathcal{V} \tag{6.4}$$

Here, d_k represents the dimension of the $\mathcal{K}$ matrix. To perform multi-branch attention, denoted as $\mathcal{B}_i$, with n branches, n sets of the *key* ($\mathcal{K}$), *query* ($\mathcal{Q}$), and *value* ($\mathcal{V}$) matrices are created using the relevant weight matrices (ω_i). Finally, a scaled dot product attention, as per Eq. 6.4, is performed on each branch as seen in Eq. 6.5.

$$\mathcal{B}_i = \alpha_{i\in[1,n]}(\mathcal{Q}_i\ \omega_i^{\mathcal{Q}}, \mathcal{K}_i\ \omega_i^{\mathcal{K}}, \mathcal{V}_i\ \omega_i^{\mathcal{V}}) \tag{6.5}$$

Next, a residual connection is applied to these tensors, followed by a layer-wise batch normalization layer. After this, a scaling factor μ is used to create the branch representation as shown below:

$$\tilde{\mathcal{B}}_i = \text{LayerNorm}(\mathcal{B}_i\omega_i^b + \mathcal{B}_i) \times \mu_i \tag{6.6}$$

In the subsequent step, a position-wise CNN (PCNN) is used on each $\tilde{\mathcal{B}}_i$ comprising of two convolution operations on each position, separated by a `ReLU` activation function. The PCNN layer operates as shown in Equation 6.7:

$$\texttt{PCNN}(x) = \texttt{Conv}(\mathit{Relu}(\texttt{Conv}(x) + b_1)) + b_2 \tag{6.7}$$

The final attentive representation of the semantic sub-spaces, generated from the PCNN layer, is obtained by carrying out a linear weighted summation (as shown in Equation 6.8), where $\gamma \in \mathbb{R}^n$ is a trainable hyper-parameter of the model.

$$\texttt{BranchAttn} = \sum_{i=1}^{n} \gamma_i \texttt{PCNN}(\tilde{\mathcal{B}}_i) \tag{6.8}$$

Finally, a residual connection is created with the output of the `BranchAttn` layer, followed by the application of a non-linear activation function (tanh). The parent node representation is calculated by performing element-wise summation (`ExS`). Equation 6.9 depicts the operation

of this step.

$$\texttt{ParentNode} = \texttt{EWS}(\tanh((\chi_{attn} + \chi)\omega + b)) \tag{6.9}$$

The attention calculation module's input and output features are χ and χ_{attn} in Eq. 6.9.

6.2.3.2 Statement Encoding Module

Once the sentence representations are generated from the sentence encoding module, the graph attention network (GAT) [224] is applied over it to generate the vector representation of the statement. For this work, we have designed the graph $\mathcal{G} = \{V, E\}$ in such a way that the sentence nodes present in the statement are connected to the statement node $\mathcal{D}$. So, for any statement comprising n sentences, there will be $n+1$ nodes in the graph (n nodes for n number of sentences and one node to represent the whole statement from the individual) and $V = \{s_1, s_2, ..., s_n, \mathcal{D}\}$. The edges are established between node $\mathcal{D}$ and the sentence nodes ($s_1, s_2, ..., s_n$), thus the graph $\mathcal{G}$ ends up with n number of edges.

This module updates only the statement node ($\mathcal{D}$) using the sentence nodes ($s_1, s_2, ..., s_n$). The sentence nodes are initialized with the sentence embeddings generated by the sentence encoding module (see Section 6.2.3.1). The GAT layer is formulated as follows:

$$\kappa_{\mathcal{D},s_j} = LeakyReLU(\omega_a[\omega_q\mathcal{D}||\omega_k s_j]) \tag{6.10}$$

$$\alpha_{\mathcal{D},s_j} = \frac{exp(\kappa_{\mathcal{D},s_j})}{\sum_{l\in\mathcal{N}_\mathcal{D}} exp(\kappa_{\mathcal{D},l})} \tag{6.11}$$

$$\mathcal{D} = \sigma(\sum_{j\in\mathcal{N}_\mathcal{D}} \alpha_{\mathcal{D},s_j}\omega_v s_j) \tag{6.12}$$

where || indicates the concatenation operation. The weight matrices ω_a, ω_q, ω_k, and ω_v in the GAT layer are updated by back-propagation. The set of neighbouring nodes for a given node is represented by $\mathcal{N}_i$, while the attention score between h_i and h_j is represented by $\alpha_{i,j}$. The GAT layer with multi-head attention, using $\mathcal{M}$ attention heads, is expressed as:

$$\mathcal{H}_i = ||_{m=1}^{\mathcal{M}}\sigma(\sum_{j\in\mathcal{N}_\mathcal{D}} \alpha^m_{\mathcal{D},s_j}\omega^m s_j) \tag{6.13}$$

This final hidden representation $\mathcal{H}_i$ is used as the statement representation vector ($\mathcal{D} = \mathcal{H}_i$).

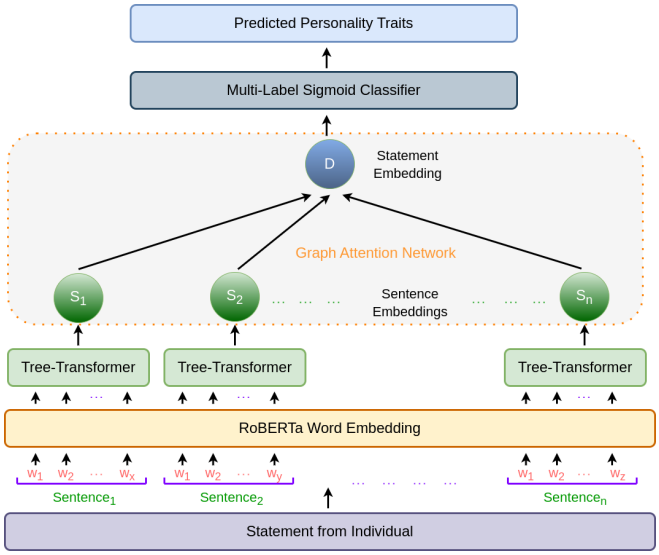

Figure 6.6: Structure of the suggested system for identifying personality traits

6.2.3.3 Model Architecture

For each individual's statement, this model at first utilizes RoBERTa [129] for generating word embeddings. In our experiments, we have also tried glove [167], fasttext [28] and BERT [56] word embeddings. However, the best results have been achieved when the model has been fed with RoBERTa word embedding.

Over these word embeddings the tree-transformers are applied to generate the sentence embeddings (see Section 6.2.3.1). The following statement encoding module then generates the embedding for the whole statement using GAT (see Section 6.2.3.2). This feature vector for the individual's statement is then fed to a dense layer with a following sigmoid classifier which returns a probability score for each personality trait. The sigmoid classifier returns the probability score for every particular traits. We have used binary cross-entropy loss function for calculating the overall loss of the model for model training. Considering N as the total number of considered personality traits, y_i as the original label and $prob(y_i)$ as the predicted probability of that particular trait, the binary cross-entropy can be formulated as:

$$loss = -\frac{1}{N}\sum_{i}^{N} y_i \log prob(y_i) + (1 - y_i) \log(1 - prob(y_i)) \tag{6.14}$$

The overall model architecture is sketched in Figure 6.6.

6.2.4 Experimental Setup and Result Analysis

In this section, we report on how well our model performed for personality trait classification, using accuracy as the evaluation metric. In the context of personality trait identification, each individual can be assigned multiple personality traits at the same time, as each trait is not mutually exclusive. Therefore, we have formulated the personality trait identification as a multi-label classification task and the performance of the model is assessed on each individual class label. This section also provides a concise overview of the benchmark datasets utilized in the experiments. We conclude this section by comparing the effectiveness of the model proposed in this study with that of the top-performing previous models.

6.2.4.1 Overview of the Benchmark Corpora

Two benchmark personality datasets: (i) Essays [166], (ii) Kaggle MBTI [99] are publicly available and have been used in our analyses.

6.2.4.1.1 Essays

The stream-of-consciousness also known as the "Essays" dataset contains 2468 essays written by students, which were annotated with binary labels over five personality traits. These binary labels indicate the presence or absence of the Big Five personality traits. These traits were identified using a standardized self-report questionnaire [166].

6.2.4.1.2 Kaggle MBTI

The data used in this corpus was accumulated from the PersonalityCafe forum, which provides a broad range of individuals interacting in an informal online social environment. The dataset consists of 8675 entries, each containing the last 50 posts made by each individual on the website. Each entry comes with its corresponding binary MBTI personality type. To work with this corpus, we have slightly modified the class labels. This corpus comes with four binary class labels: (i) Extroversion or Introversion, (ii) Sensing or Intuition, (iii) Thinking or Feeling, and (iv) Judging or Perceiving. Each entry in the corpus is labeled with four traits, one from each of four binary labels. For our experiments, we have tagged entries with 1 and 0 for Extroversion (1) and Introversion (0); Sensing (1) and Intuition (0); Thinking (1) and Feeling (0; and Judging (1) and Perceiving (0), accordingly.

6.2.4.2 Experimental Setup

The model uses an initial learning rate of 0.1 and reduces it by 80% in each iteration if the validation accuracy decreases from the previous iteration. The batch size is set to 10. The multi-branch attention block consists of six PCNN layers, and six branches of attention layer has been used for the tree-transformers in the sentence encoding module. Following Ahmed et al.'s [6] work, we have deployed each PCNN layer with two CNNs, where the first one uses 341-dimensional kernels and no dropout. The second layer utilizes 300-dimensional kernels with dropout rate 0.1. The GAT in the statement encoding unit employs six attention heads. The model hyper-parameters are trained using the 'Adagrad' [132] optimizer.

Both models use 768-dimensional RoBERTa word embeddings as input. These word embeddings are collected by feeding each sentence to the pre-trained RoBERTa. We have assessed the performance of our models using 10-fold cross-validation. To perform this cross-validation, we have utilized StratifiedKFold from the scikit-learn package. All the experiments have been conducted in a Ubuntu 22.04 LTE environment with an NVIDIA 1080ti GPU. For parsing the sentences to generate the dependency and constituency tree representations of the sentences, we have used the Stanford Core NLP parser.

6.2.4.3 Performance Analysis

Tables 6.6 and 6.7 show the accuracies achieved by our model over the two benchmark corpora along with the published results of the previous notable works. Along with the accuracies achieved over the whole corpora, accuracies over each individual class are also provided here for a better assessment of the improved results.

Looking at Table 6.6, it is clearly visible that both the proposed models outperform the previous works by a margin of 5.4 to 5.8 percentage points on average for the Essays dataset. For individual traits, the margin is 3.4 to 7.1 percentage points. The proposed models show the best performance while predicting conscientiousness. For this particular trait, the performance boost margin is 6.7 to 7.1 percentage points. And the lowest performance gain is for the label "Neuroticism" with a gain margin of 3.4 to 4.1 percentage points. In all the cases, apart from one ("Agreeableness"), the model using the dependency tree-transformer as the sentence encoding module outperforms the one which uses the constituency tree-transformer.

Table 6.7 depicts the performance of the models over the Kaggle MBTI corpus. Over Kaggle MBTI corpus, on average, our models have shown 2.9 to 3.5 percentage points performance boost compared to the previous models. Over the "Thinking/Feeling" class, the model with dependency tree-transformer has achieved 3.8 percentage points more accuracy than the previous works. The model with constituency tree's performance gain for this class is 3 percentage

Table 6.6: Performance analysis of the proposed models along with the other prominent works over the Essays dataset. All the performance scores are accuracy (in %). The best results are presented in bold texts. Here, CTT means constituency tree-transformer and DTT represents dependency tree-transformer. Column headings: O: Openness, C: Conscientiousness, E: Extraversion, A: Agreeableness, and N: Neuroticism

Model	O	C	E	A	N	Average
Previous Works						
Hierarchical CNN [136]	61.1	56.7	58.1	56.7	57.3	58.0
RNN + Mairesse [213]	58.3	63.4	59.7	57.8	60.2	59.9
BERT + Bagged SVM [105]	62.1	57.8	59.3	56.5	59.4	59.0
Psycholinguistic + MLP [142]	60.4	57.3	56.9	57.0	59.8	58.3
BERT-base + MLP [142]	64.6	59.2	60.0	58.8	60.5	60.6
BERT-large + MLP [142]	63.4	58.9	59.2	58.3	58.9	59.7
CNN-AdaBoost-2channel [151]	61.9	62.1	59.9	60.6	64.9	61.9
Proposed Models						
CTT + GAT	69.2	68.8	65.9	**65.3**	68.3	67.5
DTT + GAT	**70.1**	**69.2**	**66.5**	64.8	**69.0**	**67.9**

Table 6.7: Performance analysis of the proposed models along with the other prominent works over the Kaggle MBTI dataset. All the performance scores are accuracy (in %). The best results are presented in bold texts. Here, CTT means constituency tree-transformer and DTT represents dependency tree-transformer. Column headings: I/E: Extroversion or Introversion, S/N: Sensing or Intuition, T/F: Thinking or Feeling, and J/P: Judging or Perceiving

Model	I/E	S/N	T/F	P/J	Average
Previous Works					
BERT + Bagged SVM [105]	79.0	86.0	74.2	65.4	76.1
Psycholinguistic + MLP [142]	77.6	86.3	72.0	61.9	74.5
BERT-base + MLP [142]	78.3	86.4	74.4	64.4	75.9
BERT-large + MLP [142]	78.8	86.3	76.1	67.2	77.1
Proposed Models					
CTT + GAT	82.0	88.8	79.1	70.2	80.0
DTT + GAT	**82.5**	**89.3**	**79.9**	**70.6**	**80.6**

points over the previous works. Among the proposed models, the model with dependency tree-transformer performs better than the other one. Over all the classes, it has gained a 0.5 to 0.8 percentage point accuracy boost over the model which parses sentences using the constituency parser.

In our research, conducting an ablation study is not possible due to the interdependence of the modules in our pipeline. However, we have employed comparative studies to enhance our analysis, as presented in Table 6.8. To investigate the significance of both the tree-transformers and the GAT, we have conducted two experiments on each dataset. In the first experiment, we

Table 6.8: Comparative studies of the proposed model with different modules replaced. Row headings: RoBERTa [CLS]: The tree-transformer layer is replaced with RoBERTa CLS tokens; Mean Pooling: The GAT layer is replaced with a mean-pooling layer.

Comparison Study for the Essays Dataset						
Model	O	C	E	A	N	Average
RoBERTa [CLS]	65.3	65.1	60.8	59.2	61.7	62.4
Mean Pooling	63.8	64.1	60.2	58.3	58.9	61.1
Comparison Study for the MBTI Dataset						
Model	I/E	S/N	T/F	P/J	Average	
RoBERTa [CLS]	76.4	83.1	74.3	66.9	75.2	
Mean Pooling	75.2	81.7	71.9	65.2	73.5	

have replaced the tree-transformer layer with RoBERTa CLS tokens to generate sentence representations. In the second experiment, we have substituted the GAT layer with a mean-pooling layer over the sentence representations obtained from the tree-transformers. These experiments have allowed us to gain valuable insights into the contributions of the tree-transformer and GAT components in our model. The outcomes of our experimentation clearly indicate a decline in performance across all the aforementioned cases. When the tree-transformer layer is replaced, there is a notable drop of 5.3-5.5 percentage points (comparing averages) for the Essays dataset and 4.8-5.4 percentage points (comparing averages) for the MBTI corpus, as compared to the performance of our proposed model which can be seen in Tables 6.6 and 6.7. These findings provide compelling evidence that preserving syntactical information through tree-structured representations contributes to better semantic preservation in our model. Similar results are observed when the GAT layer is replaced with a mean-pooling layer. This time the results drop by 6.1-6.3 percentage points (comparing averages) for the Essays and 6.5-7.1 percentage points (comparing averages) for the MBTI corpora. These findings provide strong evidence that the fusion of sentence representations with attentive graph neural networks, such as GAT, can generate superior statement representations. This is attributed to the ability of GAT to assign varying weights to different sentences within a statement, despite its higher computational cost compared to a mean-pooling layer.

From these statistics, it is clear that our proposed models surpass the previous works in terms of performance. There are two reasons behind these performance boosts achieved by the proposed models. Firstly, our proposed models have the capability to work with the complete text unlike the BERT-based personality trait classifier models [105, 142]. These BERT-based models, due to the 512 word limitations of BERT, consider either only the first 512 words or the last 512 words, or the first 256 and last 256 words. On the other hand, our proposed models are able to work with sentences and texts of any length. While assessing an individual's per-

sonality traits it is important to consider that person's complete written statement. Secondly, while generating the sentence embeddings, we have utilized tree-structured representations of the sentences which has helped the models to incorporate syntactical information and preserve better semantics. Because of using dependency and constituency tree-transformers, our models can consider word-level dependencies and phrasal information. However, we have also noticed that the model with the dependency tree-transformer gives better performance compared to the model with the constituency tree-transformer. By analyzing the data, we have arrived at the hypothesis that the sentences in the benchmark corpora are reasonably simple with few phrases used and that's why considering word-level dependencies is more beneficial here. Furthermore, unlike the other models [105, 142, 213, 136], our models don't require any additional psycholinguistic features and still provide better results compared to them.

6.2.5 Conclusion

In this paper, we have proposed two models using the hierarchy of tree-transformers and graph attention network for personality trait identification and these models have outperformed the previous state-of-the-art models over Essays and Kaggle MBTI corpora. Analysis of the results also shows that using tree-structured representations while sentence embedding preserves better semantics while encoding the whole statements from individuals. Still, there are some scopes for improvement. Instead of using fixed word embeddings from BERT-based models, we can update the word embeddings like Wang et al. [227] to improve the performance of the model. Furthermore, like Kazameini et al. [106] this model can be modified to provide interpretable representations.

6.3 Detecting Personality Traits from Texts using an Hierarchy of Tree-Transformers and Graph Attention Network with Word Embedding Refinement

This section is based on the paper titled "Detecting Personality Traits from Texts using an Hierarchy of Tree-Transformers and Graph Attention Network with Word Embedding Refinement" co-authored with Robert E. Mercer. This paper is an extension of the work from Chapter 9 and is currently under review for conference publication.

Automatic detection of personality traits from individuals' written texts aids in identifying personal strengths and weaknesses, facilitating informed decisions on personal growth, workplace compatibility, and lifestyle choices. Psychologists have discerned a collection of

personality traits that can manifest within an individual's character. This research introduces a novel approach that utilizes an hierarchical structure of tree-transformers and a graph attention network (GAT) to classify personality traits derived from written text. It also employs an heterogeneous GAT (H-GAT) to refine Roberta word embeddings. The proposed model demonstrates substantial performance enhancements compared to previous works, as evidenced by superior results on benchmark datasets.

6.3.1 Introduction

Personality refers to the enduring traits and patterns of behavior that an individual consistently displays. It encompasses a person's moods, attitudes, and opinions, which are explicitly manifested in their interactions with others. Personality encompasses a wide range of behavioral characteristics, both innate and acquired, that are observable in an individual's social relationships and their interactions with the surrounding environment. These traits significantly influence an individual's future prospects and life outcomes [185].

The Big-Five personality traits, also known as OCEAN, are the widely accepted and commonly used model of personality [97]. OCEAN represents personality through five dimensions: Openness to Experience, Conscientiousness, Extraversion, Agreeableness, and Neuroticism [106]. Another frequently employed personality model is the Myers-Briggs Type Indicator (MBTI) [153], which categorizes individuals into 16 distinct personality types based on four binary categories: Extroversion or Introversion, Sensing or Intuition, Thinking or Feeling, and Judging or Perceiving.

Extensive studies, most notably on the writings of freshmen university students, have promoted the investigation of language to determine personality traits [166]. Researchers, such as Kosinski et al. [111], have shown that extensive social media footprints can provide insights into an individual's personality and emotional traits.

Many prevailing and contemporary cutting-edge models for classifying personality traits revolve around the utilization of BERT-based architectures [143, 105]. These language-based models have been applied to the Essays dataset [166] and online posts. Ramezani et al. [179] develops an attention-based method that uses various knowledge graphs to classify personality traits in the Essays dataset. Yang et al. [248], Zhu et al. [274], and Yang et al. [247] design graph-based models using psychological and semantic relations among posts.

In this study, we present a novel language-based approach to classify the personality traits that can be drawn from text. We propose three distinct models that incorporate tree-transformers [6], a graph attention network (GAT) [224], and an heterogeneous graph attention network (H-GAT) [227]. Each of these architectures employs an hierarchical structure com-

prising tree-transformers and GAT layers. The tree-transformers serve as sentence encoders, while the subsequent GAT layer encodes complete statements using the derived sentence vectors. To update the leaf nodes of the tree-transformers and sentence nodes, an H-GAT has been deployed, which leverages the statement embedding. Notably, the three models vary in the specific application of the H-GAT. By fine-tuning the word embeddings, these models effectively serve the purpose of BERT fine-tuning. Their advantage is reducing the need for substantial computational resources to fine-tune the millions of parameters found in the BERT-based models and enabling essentially unlimited input text lengths. In our study, we have conducted an extensive analysis of the performance of the proposed models on well-established personality trait identification datasets. Through rigorous analysis, the findings unequivocally show the superior performance of our proposed model when compared to previously prominent models in the field.

6.3.2 Related Work

In recent years, there have been notable contributions in employing various deep learning models for the identification of personality traits. Su et al. [212] utilized recurrent neural networks (RNN) along with hidden Markov models (HMM) to identify personality traits using Chinese Language Inquiry and Word Count (LIWC) annotations extracted from dialogues. Sun et al. [213] and Thandera et al. [215] utilized long-short-term-memory (LSTM) and convolutional neural networks (CNN) to detect personality traits from text data sourced from Facebook posts. Van et al. [221] have conducted experiments on 275 LinkedIn profiles and provided evidence that extroversion can be accurately inferred from self-descriptions in user profiles. Lynn et al. [133] employed message-level attention over Facebook posts to analyze users' personality traits. Gjurkovic et al. [74] have introduced their self-created corpus in the context of personality analysis and applied S-BERT [182] over it. Kazameini et al. [105] utilized an ensemble of SVMs with BERT embeddings and achieved superior performance compared to other models for the Big Five trait classification using the Essays corpus [166]. Mehta et al. [142] have conducted experiments with various combinations of psycholinguistic features and BERT-based models, analyzing the impact of each feature on trait prediction. Stachl et al. [211] and Mehta et al. [143] delve into computational perspectives in their review article, exploring various aspects and considerations within the field of personality trait identification. Ramezani et al. [179] have incorporated a knowledge graph with CNN, RNN, LSTM and Bi-LSTM for automatic personality trait classification. Yang et al. [248] uses a graph attention network (GAT) [224] approach that utilizes LIWC [217]. Zhu et al. [274] generates two graphs, one linking posts if they contain words in the same LIWC categories and one representing semantically

similar posts, which are used in a contrastive graph transformer network (CGTN). Yang et al. [247] builds a dynamic deep graph convolutional network (D-DGCN) incorporating information about each individual's posts. These graph-based methods use pre-trained BERT embeddings. In addition, the methods are applicable to online posts only, as they are based on multiple posts by an individual.

6.3.3 Methodology

The personality trait classification model utilizes an hierarchical framework with a sentence encoder and a statement encoder. The sentence encoder generates vectors for each sentence, while the statement encoder synthesizes these vectors for the entire text. We have experimented with two tree-transformer variants as sentence encoders: constituency (CTT) and dependency (DTT) tree-transformers. A graph attention network (GAT) merges sentence representations and produces a statement representation. The heterogenous GAT (H-GAT) layer refines sentence and word nodes using the statement vector. Three model architectures have been examined, varying the configuration of the H-GAT layer. It enhances sentence and word representation by incorporating information from the statement vector. This section explains the individual components and then describes the comprehensive model.

6.3.3.1 Sentence Encoder Module

To effectively analyze an individual's personality traits through textual data, we take into account the syntactical structure of sentences. Motivated by Tai et al. [214], we address this requirement by exploring two types of tree-structured transformer models which capture correlations between distant words and the phrasal structures present in sentences. While attention mechanisms [18, 223] have made significant strides in addressing the issue of long-distance dependencies, they still fall short when compared to tree-structured models [5, 6].

To convey comprehensive information about a sentence, two types of tree-based representations are employed: constituency trees, which capture distinct aspects of sentence syntax, and dependency trees, which achieve the relationships between individual words positioned at various locations within the sentence [6]. Through recursive computations entailing attention across branches, the models analyze each sub-tree and generate a sentence vector representation at the tree's root.

To generate the self-attention over the branches query ($\mathcal{Q}$), key ($\mathcal{K}$), and value ($\mathcal{V}$) matrices

are computed [223] (see Eqs. 6.15-6.17).

$$\mathcal{K} = \omega_k \mathcal{M}_k \quad \text{s.t.} \quad \omega_k \in \mathbb{R}^{d \times d} \tag{6.15}$$

$$\mathcal{V} = \omega_v \mathcal{M}_v \quad \text{s.t.} \quad \omega_v \in \mathbb{R}^{d \times d} \tag{6.16}$$

$$\mathcal{Q} = \omega_q \mathcal{M}_q \quad \text{s.t.} \quad \omega_q \in \mathbb{R}^{d \times d} \tag{6.17}$$

For a DTT, the matrix $\mathcal{M}$ is constructed by concatenating the word vectors of all child nodes associated with each parent node. Conversely, in a CTT, the matrix $\mathcal{M}$ is formed by concatenating the word vectors within a constituent. The self-attention matrix (α) is calculated by leveraging the $\mathcal{Q}, \mathcal{V}, \mathcal{K}$ matrices in the following manner:

$$\alpha = \text{softmax}(\frac{\mathcal{Q}\,\mathcal{K}^T}{\sqrt{d_k}})\mathcal{V} \tag{6.18}$$

Here, the dimension of the key ($\mathcal{K}$) matrix is denoted as d_k.

To carry out multi-branch attention, denoted as $\mathcal{B}_i$, with n branches, n sets of the key ($\mathcal{K}$), query ($\mathcal{Q}$), and value ($\mathcal{V}$) matrices with n corresponding weight matrices (ω_i) are used. Next, a scaled dot product attention is performed on each branch:

$$\mathcal{B}_i = \alpha_{i \in [1,n]}(\mathcal{Q}_i\ \omega_i^{\mathcal{Q}}, \mathcal{K}_i\ \omega_i^{\mathcal{K}}, \mathcal{V}_i\ \omega_i^{\mathcal{V}}) \tag{6.19}$$

Next, a residual connection is introduced to the tensors obtained from the multi-branch attention operation followed by a layer-wise batch normalization layer to normalize the tensor outputs and finally a scaling factor μ is used:

$$\tilde{\mathcal{B}}_i = \text{LayerNorm}(\mathcal{B}_i \omega_i^b + \mathcal{B}_i) \times \mu_i \tag{6.20}$$

In the subsequent stage, a position-wise CNN (PCNN) is applied to each $\tilde{\mathcal{B}}_i$. The PCNN layer consists of two convolution operations performed at each position, with a ReLU activation function separating the convolution operations:

$$\texttt{PCNN}(x) = \texttt{Conv}(\mathit{ReLU}(\texttt{Conv}(x) + b_1)) + b_2 \tag{6.21}$$

Then, the attentive representation of the semantic sub-spaces is generated by applying a linear weighted summation over the PCNN layer-derived features (see Eq. 6.22). Here γ is a trainable hyper-parameter of the model that determines the weights assigned to each semantic

sub-space.

$$\texttt{BranchAttn} = \sum_{i=1}^{n} \gamma_i \texttt{PCNN}(\tilde{\mathcal{B}}_i) \tag{6.22}$$

Finally, a residual connection is established between the output of the BranchAttn layer and the subsequent step. Then, a non-linear activation function (tanh) is applied to the resulting tensor. The parent node representation is then computed by performing an element-wise summation (EwS) which combines the representations of the child nodes:

$$\texttt{ParentNode} = \texttt{EWS}(\tanh((\chi_{att} + \chi)\omega + b)) \tag{6.23}$$

Here, the input features to the attention calculation module are denoted as χ, while the output features are represented as χ_{att}.

To incorporate both the word-level dependencies and the underlying phrasal information present in the sentences, mean pooling is utilized over the sentence vectors obtained from the DTT and CTT. This is elaborated in Section 6.3.3.4.

6.3.3.2 Statement Encoder Module

Over the sentence representations generated from the sentence encoding module (see Section 6.3.3.1), the Graph Attention Network (GAT) [224] is employed to generate the vector representation of the statement. The graph $\mathcal{G} = \{V, E\}$ is designed in such a way that there is an edge between the statement node $\mathcal{D}$ and all n sentence nodes $(S_1, S_2, ..., S_n)$ in the statement. Thus, $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ where $\mathcal{V} = \{S_1, S_2, ..., S_n, \mathcal{D}\}$ and $\mathcal{E} = \{S_1 \rightarrow \mathcal{D}, S_2 \rightarrow \mathcal{D}, ..., S_n \rightarrow \mathcal{D}\}$. The sentence nodes are initialized with the sentence embeddings that are generated by the sentence encoder module and by applying mean pooling over them, $\mathcal{D}$ is initialized. GAT is applied over these sentence nodes to generate the vector representation for node $\mathcal{D}$ (see Eqs. 6.24-6.26).

$$\kappa_{\mathcal{D},S_j} = \texttt{LeakyReLU}(\omega_a[\omega_q\mathcal{D}||\omega_k S_j]) \tag{6.24}$$

$$\alpha_{\mathcal{D},S_j} = \frac{\exp(\kappa_{\mathcal{D},S_j})}{\sum_{l\in\mathcal{N}_\mathcal{D}} \exp(\kappa_{\mathcal{D},l})} \tag{6.25}$$

$$\mathcal{D} = \sigma(\sum_{j\in\mathcal{N}_\mathcal{D}} \alpha_{\mathcal{D},S_j}\omega_v S_j) \tag{6.26}$$

Here, the concatenation operation is denoted by $||$. ω_a, ω_q, ω_k, and ω_v are the trainable weight matrices. The set of neighbouring nodes for a given node (S_i or $\mathcal{D}$) is represented by $\mathcal{N}_i$. $\alpha_{i,j}$ denotes the attention value between any two nodes in the graph. The GAT layer incorporates multi-head attention. Utilizing $\mathcal{M}$ attention heads, this multi-head attention formulation can be

expressed as follows:

$$\mathcal{H}_i = \|_{m=1}^{\mathcal{M}} \sigma(\sum_{j \in \mathcal{N}_{\mathcal{D}}} \alpha_{\mathcal{D},S_j}^{m} \omega^m S_j) \tag{6.27}$$

This final hidden representation $\mathcal{H}_i$ is used as the statement representation vector ($\mathcal{D} = \mathcal{H}_i$).

6.3.3.3 Refinement module

The refinement module employs the heterogeneous graph attention network (H-GAT) to update the word and sentence embeddings based on the statement embeddings generated from the statement update module. This refinement module is inspired by the H-GAT [227]. Originally designed to enhance cross-sentence relationships and to generate more informative sentence representations for extractive summarization tasks, we have adapted this approach to improve the quality of statement representations for our task. In our methodology, the H-GAT module is utilized at each iteration, following the completion of forward passes of the sentence encoder and statement encoder modules. By incorporating the statement-to-sentence, sentence-to-word, and statement-to-word update steps and subsequent forward passes of the sentence and statement encoder modules, this module enriches the statement vectors, leading to an enhancement in the overall quality of the statement representations for the personality trait detection task. This section outlines the general concept of the refinement module. The varying placements of this module are elaborated in Section 6.3.3.4.

Considering a statement has n sentences, the **statement-to-sentence update module** works on a graph $G = \{V, E\}$ where the set of vertices $V = \{S_1, S_2, ..., \mathcal{D}\}$ and set of edges $E = \{S_1 \leftarrow \mathcal{D}, S_2 \leftarrow \mathcal{D}, ..., S_n \leftarrow \mathcal{D}\}$ (similar to the graph $\mathcal{G}$ in the statement encoding module). After constructing the graph G, the feature values of the nodes are modified using a Graph Attention Network (GAT) [224]. Let $h_i \in \mathbb{R}^{d_h}$ represent the hidden states of the statement and sentence nodes, where $i \in \{1, 2, ..., (n+1)\}$, and d_h denotes the dimension of the hidden states. The GAT layer, which operates on this graph, can be formulated as follows:

$$\kappa_{i,j} = \texttt{LeakyReLU}(\omega_a[\omega_q h_i; \omega_k h_j]) \tag{6.28}$$

$$\alpha_{i,j} = \frac{\exp(\kappa_{i,j})}{\sum_{l \in \mathcal{N}_i} \exp(\kappa_{i,l})} \tag{6.29}$$

$$\mathcal{Z}_i = \sigma(\sum_{j \in \mathcal{N}_i} \alpha_{i,j} \omega_v h_j) \tag{6.30}$$

where ω_a, ω_q, ω_k, and ω_v are the weight matrices in the GAT layer are updated during the back-propagation based on the gradients. The set of neighbouring nodes for a given node i

is represented by $\mathcal{N}_i$, and the attention score between hidden states h_i and h_j is denoted as $\alpha_{i,j}$. This layer acts similarly to the GAT layer explained in Eqs. 6.24-6.26. But this time, it updates the sentence nodes based on the statement node and updates the word nodes based on the sentence nodes.

To enhance the expressiveness of the GAT layer, it can be extended to incorporate multi-head attention with $\mathcal{M}$ heads. This extension allows the model to capture multiple aspects or perspectives of the relationship between nodes. The formulation of the GAT layer with multi-head attention can be expressed as follows:

$$\mathcal{Z}^i = \|_{m=1}^{\mathcal{M}} \sigma(\sum_{j \in \mathcal{N}_i} \alpha_{i,j}^m \omega^m h_i) \tag{6.31}$$

Finally, a residual connection is established in the model. This connection allows the final hidden state representation h_i to incorporate the information u_i from the residual connection. The updated hidden state representation is formulated as $h_i = u_i + h_i$. This addition operation ensures that the information from previous layers is preserved and combined with the current representation, helping to alleviate the issue of vanishing gradients.

At each iteration, the sentence nodes undergo updates using the GAT layer and a position-wise feed-forward network (FFN) layer. Following the approach introduced by Wang et al. [227], the updates are performed considering the information from the statement node. The updates can be described by the following equations:

$$\mathcal{Z}_{\mathcal{D}\rightarrow S}^{t+1} = GAT(\mathcal{H}_S^t, \mathcal{H}_{\mathcal{D}}^t, \mathcal{H}_{\mathcal{D}}^t) \tag{6.32}$$

$$\mathcal{H}_S^{t+1} = FFN(\mathcal{Z}_{\mathcal{D}\rightarrow S}^{t+1} + \mathcal{H}_{\mathcal{D}}^t) \tag{6.33}$$

In Eq. 6.32, at the first iteration ($t = 0$), $\mathcal{H}_S^0$ corresponds to the initial set of sentence nodes, which are obtained from the sentence encoder module. On the other hand, $\mathcal{H}_{\mathcal{D}}^0$ represents the statement representation derived from the statement encoder module. In the GAT layer, $\mathcal{H}_S^t$ serves as the query matrix, while $\mathcal{H}_{\mathcal{D}}^t$ is utilized as both the value and key matrices. This configuration is inspired by the approach in Vaswani et al. [223], aiming to capture the attention-based relationships between the sentence nodes and the statement representation.

Both the sentence-to-word and statement-to-word update steps are designed following the same principle of the statement-to-sentence update step. The **sentence-to-word** update step tries to refine the word embeddings based on the sentence embedding so that the word vectors can preserve the essence of the sentence. For any sentence S containing p words, the word

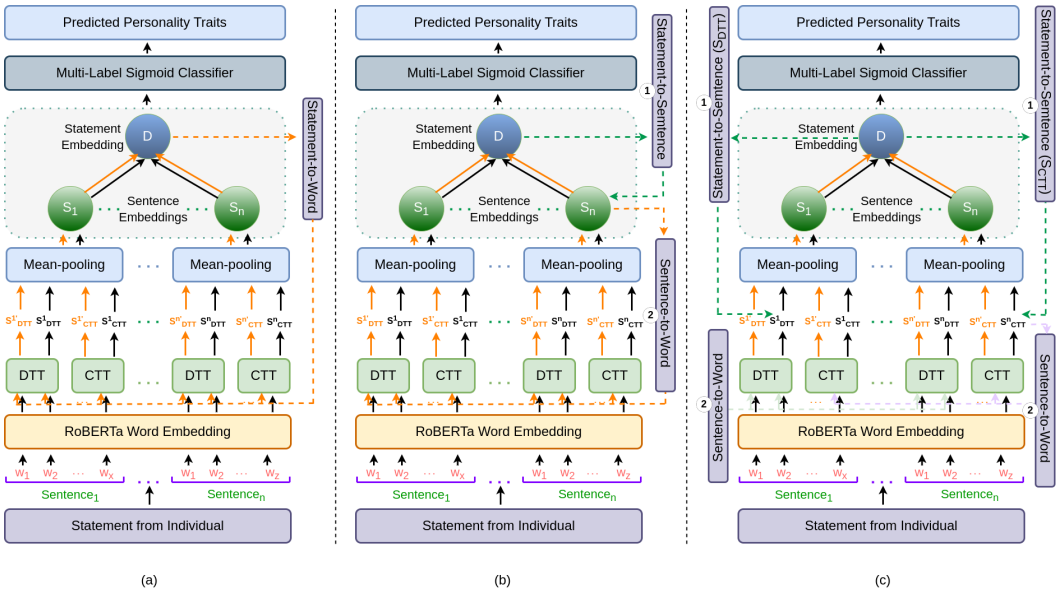

Figure 6.7: Structure of the investigated systems for identifying personality traits. (a) the word embeddings are updated using statement vector, (b) the statement vector updates the S_{avg} and subsequently updates the word embeddings, and (c) the statement vector updates the S_{DTT} and S_{CTT}. These sentence vector updates the word embeddings separately. All the unbroken orange straight lines indicate the second forward pass with the updated word vectors. For (b) and (c) the refinement steps labeled with numbers indicate the order of occurrence.

nodes are updated by a GAT layer as follows:

$$\mathcal{Z}^{t+1}_{S\rightarrow w} = GAT(\mathcal{H}^t_w, \mathcal{H}^t_S, \mathcal{H}^t_S) \tag{6.34}$$

$$\mathcal{H}^{t+1}_w = FFN(\mathcal{Z}^{t+1}_{S\rightarrow w} + \mathcal{H}^t_S) \tag{6.35}$$

where, at the first epoch ($t = 0$), $\mathcal{H}^0_w$ represents the initial set of word nodes. These word nodes correspond to the RoBERTa-based embeddings [129] for the words in the sentence. $\mathcal{H}^t_S$ depicts the updated sentence representations obtained from the statement-to-sentence update step.

The statement-to-word update step applies GAT to produce refined word embeddings with the knowledge of the statement embedding. For the statement $\mathcal{D}$, this step is defined as:

$$\mathcal{Z}^{t+1}_{\mathcal{D}\rightarrow w} = GAT(\mathcal{H}^t_w, \mathcal{H}^t_{\mathcal{D}}, \mathcal{H}^t_{\mathcal{D}}) \tag{6.36}$$

$$\mathcal{H}^{t+1}_w = FFN(\mathcal{Z}^{t+1}_{\mathcal{D}\rightarrow w} + \mathcal{H}^t_{\mathcal{D}}) \tag{6.37}$$

where, initially ($t = 0$), $\mathcal{H}^0_w$ is the set of word nodes present in the statement $\mathcal{D}$ and initialized with the RoBERTa word embeddings. $\mathcal{H}^0_{\mathcal{D}}$ is the statement vector generated by the statement encoder.

6.3.3.4 Model Architecture

By varying the position and utilization of the refinement module units, we have investigated three architectures for the automatic personality trait detection task. The architectural structures of the proposed models are portrayed in Figure 6.7. In the context of the personality trait detection task, all of the models require two forward passes separated by a refinement step. The first forward pass is a common step shared by all of the models. During the initial forward pass, RoBERTa word embeddings are used as the initial input to the model. Subsequently, the aforementioned inputs undergo simultaneous processing by both the DTTs and CTTs in the sentence encoder module. This step outputs two sentence representations for each sentence in the statement: $S_{DTT} \in \{S^1_{DTT}, S^2_{DTT}, ..., S^n_{DTT}\}$ and $S_{CTT} \in \{S^1_{CTT}, S^2_{CTT}, ..., S^n_{CTT}\}$, accordingly. Following this stage, a mean-pooling procedure is executed, resulting in the generation of an intermediate sentence representation denoted as S_{avg}. Thus for a statement D containing n sentences, n sentence representations ($S_{avg} \in \{S^1_{avg}, S^2_{avg}, ..., S^n_{avg}\}$) are generated. These sentence representations from the sentence encoder are passed to the statement encoder module. The GAT layer in the statement encoder computes the statement representation $\mathcal{D}$ and the first forward pass ends here.

The major difference between the investigated models is the utilization and design of the refinement step. For the first model (see Figure 6.7(a)), the refinement module uses only the statement-to-word update step. In the second investigated model (see Figure 6.7(b)), the refinement module uses the statement-to-sentence and the sentence-to-word update steps. The statement-to-sentence step, at first, updates the averaged sentence representations (S_{avg}). These updated sentence representations are then used by the sentence-to-word update module to update the word embeddings. The last model (see Figure 6.7(c)) also uses the statement-to-sentence and the sentence-to-word update steps. But here, the statement-to-sentence update module updates the S_{DTT} (S'_{DTT}) and S_{CTT} (S'_{CTT}). Then, the sentence-to-word refinement step is utilized twice: once to update the word embeddings based on the updated S_{DTT}, and another time based on the updated S_{CTT}.

After the refinement module is employed, the second forward pass is initiated. For the first two models, with the updated word embeddings, the forward pass is the same as the first forward pass. But for the third model, the sentence encoder module works with two different word embeddings. The CTT intakes the word embeddings updated by the S'_{CTT}, and the DTT is fed with word embeddings updated by the S'_{DTT} as inputs. The following steps are similar to the other two models. This second forward pass generates a refined statement vector ($\mathcal{D}'$). Subsequently, $\mathcal{D}'$ is fed into a dense layer, followed by a *sigmoid* classifier that assigns a probability score to each individual personality trait. For model training, we have employed

the binary cross-entropy loss function to evaluate and calculate the overall loss of the model.

6.3.4 Experimental Setup

Here, we give an assessment of our model's efficacy in discerning personality traits, employing F1 score and macro-F1 score as the evaluation metrics. Individuals can exhibit multiple traits concurrently, given that these characteristics are not inherently exclusive. Consequently, we have framed the identification of personality traits as a multi-label classification problem, gauging the model's performance against each distinct class label. Furthermore, this section provides a synopsis of the benchmark datasets used in our experiments.

We have experimented on three publicly available benchmark corpora: (i) Essays [166], (ii) Kaggle [99], and (iii) Pandora [74]. The Essays dataset encompasses a collection of 2468 compositions penned by students, meticulously annotated with binary labels pertaining to five distinct personality traits: Openness (O), Conscientiousness (C), Extraversion (E), Agreeableness (A), and Neuroticism (N). They were annotated by analyzing a standardized self-report questionnaire for each student. Each entry in Kaggle and Pandora is associated with a binary MBTI personality type. These corpora encompass four binary class labels, namely: (i) Extroversion or Introversion (I/E), (ii) Sensing or Intuition (S/N), (iii) Thinking or Feeling (T/F), and (iv) Judging or Perceiving (J/P). The Kaggle and Pandora datasets comprise a substantial collection of 8675 and 9067 records, respectively. Each entry in Kaggle contains the 50 most recent contributions made by individuals on the PersonalityCafe website, whereas the records in Pandora are collected from Reddit. The data pre-processing step follows the approach used in (to preserve anonymity we don't cite the work. Upon acceptance we will add the citation.). The statistics of the corpora are presented in Appendix 6.3.8.2. For all three corpora, we have performed 10-fold cross-validation with a split of 80/10/10 of the samples for the training/validation/testing.

The model employs an initial learning rate of 0.1, which is subsequently reduced by 80% in each iteration if the validation accuracy declines compared to the previous iteration. The batch size is 10. For the tree-transformers, the same hyper-parameter settings are used as in Ahmed et al. [6]. The statement encoding unit utilizes a GAT (Graph Attention Network) with six attention heads. The model's parameters are trained using the "Adagrad" optimizer [132].

The output representations for the sentence encoders (DTT and CTT), the statement encoder, and the model itself, are 768-dimensional vectors. The model employs two forward passes to generate the statement vector. During the first forward pass, RoBERTa word embeddings are utilized. In the second pass, the updated word representations obtained from the "refinement module" are employed, as described in Section 6.3.3.4. The performance eval-

Table 6.9: Performance analysis of the proposed models along with the other prominent works over the Essays dataset. The reported results are the F1 scores. The best results are presented in bold texts.

	F1 Score					
Model	O	C	E	A	N	Ave.
Previous Works						
Psycholinguistic + MLP [142]	61.11	57.68	57.72	58.46	59.79	58.95
BERT-large + MLP [142]	64.09	59.27	60.01	59.75	58.49	60.32
CNN-AdaBoost-2channel [151]	62.60	62.46	60.71	62.02	64.89	62.54
KGrAT-Net [179]	75.04	76.52	78.14	72.83	69.91	74.43
$CGTN_{pretrain}$ [274]	72.28	74.75	76.21	76.01	73.77	74.60
$CGTN_{joint}$ [274]	72.17	76.21	78.78	77.12	70.87	75.03
Proposed Models						
Model-1	72.50	73.50	74.44	72.22	70.59	72.65
Model-2	75.45	76.08	77.79	75.04	73.09	75.49
Model-3	**76.27**	**78.71**	**80.29**	**78.78**	**77.59**	**78.33**

Table 6.10: Performance analysis of the proposed models along with the other prominent works over the Kaggle and Pandora MBTI datasets. The reported results are the macro-F1 scores. The best results are presented in bold texts. Missing values are presented with -.

	Kaggle F1 Score					Pandora F1 Score				
Model	I/E	S/N	T/F	P/J	Ave.	I/E	S/N	T/F	P/J	Ave.
Previous Works										
Psycholinguistic + MLP [142]	72.84	77.52	71.90	61.25	70.88	-	-	-	-	-
BERT-large + MLP [142]	74.13	77.52	76.00	66.54	73.55	-	-	-	-	-
TrigNet [248]	69.54	67.17	79.06	67.69	70.86	56.69	55.57	66.38	57.27	58.98
D-DGCN and D-DGCN-ℓ_0 [247]	69.52	67.19	80.53	68.16	71.35	61.55	55.46	71.07	59.96	62.49
Proposed Models										
Model-1	80.21	83.20	83.12	72.36	79.72	70.12	72.13	73.29	64.84	70.10
Model-2	80.66	87.51	85.72	75.48	82.34	71.08	78.15	76.60	68.75	73.50
Model-3	**81.35**	**88.65**	**86.63**	**76.08**	**83.18**	**71.92**	**79.23**	**77.56**	**69.35**	**74.52**

uation of our models has been conducted using 10-fold cross-validation. To facilitate this cross-validation process, we have utilized the StratifiedKFold function from the scikit-learn package. All experiments have been conducted in an Ubuntu 22.04 LTE environment, leveraging a 48GB NVIDIA RTX A6000 GPU. For parsing the sentences and generating the tree representations, we have used the Stanford Core-NLP parser [137].

6.3.5 Analysis of Results

Tables 6.9 and 6.10 showcase the performance of the proposed models on the Essays and the Kaggle and Pandora corpora, respectively. The results clearly demonstrate that our proposed

models have outperformed previous models, including the current state-of-the-art (SOTA) models [179, 274, 142, 247], by a significant margin without using any additional features like the other models do. For the Essays corpus, the model incorporating the statement-to-word update module (Model-1) exhibits slightly lower F1 scores compared to the current SOTA [179, 274]. The second model, which incorporates the statement-to-sentence (S_{avg}) and sentence (S_{avg})-to-word refinements, approximates the F1 scores of the current SOTA [179, 274]. The third model, which incorporates separate statement-to-sentence (S_{DTT} and S_{CTT}) and sentence (S_{DTT} and S_{CTT})-to-word update modules, exhibits better class scores (class N showing the most improvement) and a 3.3 average F1 point improvement over the SOTA. A similar performance boost is observed in the experiments with the Kaggle corpus, as well. Our best performing proposed model (Model-3) has shown 9.63 point gain, on average, over the BERT-large + MLP model which is the current SOTA. With the Pandora corpus, the best performing model is the D-DGCN-ℓ_0 achieving an average 62.49 macro-F1 score. All of our proposed models outperform it by achieving 7.61 to 12.03 point higher average macro-F1 scores.

KGrAT-Net incorporates the knowledge graph to improve the performance of the model, whereas CGTN links posts by determining the common words that are in the same LIWC category. D-DGCN also tries to generate a graph of the essays from the individuals post. These models try to identify personality traits based on multiple posts from each individual. However, they ignore the linguistic features of these posts which in our work we have tried to accommodate. Incorporation of the linguistic features (parse tree-structures) allow our model to understand the texts better and analyze individual posts regarding personality traits.

We believe the reason behind the improvement over the BERT-based models is that the BERT-based models [142] work with only the first 512 tokens of the statements due to the token input limitation of BERT. Our model has surpassed that limitation by using the tree-transformer based sentence encoder module. It works with individual sentences from the statement and thus it is not dependent on the statement length. Furthermore, the statement encoder module imposes attention over the sentences which helps the model understand which sentences are important when identifying personality traits.

Another reason for the improved performance is that the other models use the features from pre-trained BERT models without any fine-tuning for this task whereas our proposed models, using the refinement module, update the word embeddings as well, based on the generated statement representation which in the end helps to produce more enriched statement representations. This approach is quite similar to the concept of fine-tuning BERT-based models, but demands less computational resources (122M vs 345M parameters) and one-fourth of the training time compared to the BERT-fine-tuning, making our model more suitable to run on computers with less computational resources. The proposed model takes slightly more time

compared to the BERT-large model in the testing phase. However, there are ways to parallelize our model to reduce this computational time difference.

Using the tree-transformers allows the model to better capture structural knowledge at the sentence level. Through our experiment, we have found that while dealing with complex sentences the other models fail to identify all of the personality traits properly as there exist dependencies between different phrases at various distances in the sentence.

Among the proposed models, we observe that the second and third models perform much better than the first one. The second model directly refines the word representations based on the generated statement vector. It requires a lot of nodes to be refined all together based on the value of only one node (the statement vector) and the refinement ignores the sentence-level information. The third model uses two separate statement-to-sentence and sentence-to-word update modules so the two tree-transformers get different word embeddings and allows the model to have more semantic information during the second pass, helping it to achieve higher performance compared to the second model. To show the importance of the individual components of these methods, an ablation study is shown in Appendix 6.3.8.1. The ablation study demonstrates that the refinement module helps the model to achieve superior performance. From the ablation study, we have observed that, across all three corpora, the refinement step helps the model to improve the performance by 4 to 9 F-1 score points. However, our model makes wrong predictions in some cases. Each class within the Essays corpus is exemplified by a singular case in Appendix 6.3.8.3.

6.3.6 Conclusions and Future Work

This study introduces three innovative architectures that leverage an hierarchical structure of tree-transformers and a graph attention network for the classification of personality traits inferred from written text. The refinement module proposed in this research aids in the precise adjustment of word vectors while preserving enriched semantics and syntactical information. The proposed models have demonstrated a substantial performance improvement compared to previous prominent works. A potential extension of this work could involve the incorporation of a knowledge graph, similar to the approach taken by Ramezani et al. [179].

6.3.7 Limitations

In this study, our focus was primarily on the Big Five Model (OCEAN) and Myers-Briggs Type Indicator (MBTI) personality trait classifications. However, it is important to note that there are two other noteworthy personality trait models that warrant attention: Eysenck's Personality Dimensions and the HEXACO Model. These alternative models offer distinct frameworks for

understanding and categorizing personality traits. We are not sure how well these models will perform when working with them.

Moreover, while the proposed models have indeed exhibited a substantial performance improvement, it is important to acknowledge that there is a trade-off in terms of computational time. In the first forward pass the model takes the RoBERTa word embeddings initially and then generates the sentence and statement representations. From this generated statement and sentence representations, the word representations are updated. In the second forward pass, using the context-enriched word embeddings the sentence and statement representations are generated again and the personality traits are identified. Using two forward passes plus parsing required for the tree-structured transformers in the model leads to an increase in time required for the generation of results compared to the other models. This computational overhead should be taken into consideration when considering the deployment and scalability of the proposed models in practical applications. However, with some parallelization in the model implementation, the computational time it requires can be reduced.

6.3.8 Appendix

6.3.8.1 Ablation Study

Table 6.11: Ablation Study on the Essays dataset. Here, CTT + GAT is the model where sentences are encoded with only the constituency tree-transformer (CTT) and only the graph attention network (GAT) generates the statement encoding. No refinement module is used. DTT + GAT uses the dependency tree-transformer (DTT) as the sentence encoder and GAT as the statement encoder without any refinement module. DTT + CTT + GAT takes the point-wise average of the sentence representations generated from the DTT and CTT, and the GAT layer computes the statement vector. No refinement module is used here, as well. All the performances are macro-F1 scores (in %).

Model	O	C	E	A	N	Average
CTT + GAT	67.03	66.18	65.55	65.14	66.76	66.13
DTT + GAT	67.54	66.70	66.09	65.72	67.34	66.68
DTT + CTT + GAT	68.76	68.04	67.01	66.71	68.14	67.73

Observing the results in Tables 6.9 and 6.11, Tables 6.10 (Kaggle) and 6.12, and Tables 6.10 (Pandora) and 6.13, we can clearly say that the performance of the individual units are much lower compared to the proposed models. The refinement unit, present in the proposed models, plays the vital role in the performance boost achieved by the three investigated models.

Table 6.12: Ablation Study on the Kaggle dataset. Here, CTT + GAT is the model where sentences are encoded with only the constituency tree-transformer (CTT) and only the graph attention network (GAT) generates the statement encoding. No refinement module is used. DTT + GAT uses the dependency tree-transformer (DTT) as the sentence encoder and GAT as the statement encoder without any refinement module. DTT + CTT + GAT takes the point-wise average of the sentence representations generated from the DTT and CTT, and the GAT layer computes the statement vector. No refinement module is used here, as well. All the performances are macro-F1 scores (in %)

Model	I/E	S/I	T/F	P/J	Average
CTT + GAT	75.39	79.15	78.92	68.19	75.41
DTT + GAT	76.20	80.51	79.25	69.16	76.28
DTT + CTT + GAT	77.82	81.02	79.98	71.07	77.48

Table 6.13: Ablation Study on the Pandora dataset. Here, CTT + GAT is the model where sentences are encoded with only the constituency tree-transformer (CTT) and only the graph attention network (GAT) generates the statement encoding. No refinement module is used. DTT + GAT uses the dependency tree-transformer (DTT) as the sentence encoder and GAT as the statement encoder without any refinement module. DTT + CTT + GAT takes the point-wise average of the sentence representations generated from the DTT and CTT, and the GAT layer computes the statement vector. No refinement module is used here, as well. All the performances are macro-F1 scores (in %)

Model	I/E	S/I	T/F	P/J	Average
CTT + GAT	69.48	71.33	72.56	64.82	69.55
DTT + GAT	70.10	71.89	73.01	64.78	69.95
DTT + CTT + GAT	71.06	71.23	73.88	65.62	70.45

6.3.8.2 Statistics of the Corpora

6.3.8.2.1 Statistics of the Essays Corpus

Table 6.14: Statistics of the Essays dataset.

	Openness	**Conscientiousness**	**Extraversion**	**Agreeableness**	**Neuroticism**
Positive	1271	1253	1276	1310	1233
Negative	1197	1215	1192	1158	1235

6.3.8.2.2 Statistics of the MBTI Corpora

Table 6.15: Statistics of the MBTI datasets.

	Kaggle	Pandora
Category	**Number of Samples**	**Number of Samples**
Extroversion	1999	1925
Introversion	6677	7142
Sensing	1197	1445
Intuition	7479	7922
Thinking	3981	5851
Feeling	4695	3216
Judging	3434	3757
Perceiving	5242	5310

6.3.8.3 Errors in Prediction Made by the Proposed Model

6.3.8.3.1 Openness

I just got back from your class, so I decided that I should start to type this paper. I am very happy with my classes, even though I feel like they are going to be rather difficult this year, especially my Calculus class. I have a hard time understanding what my professor is saying. I end up have to go home and teach myself most of the information. Well that's enough about school. I just thought about my exgirlfriend. I have very strong emotions about her. I know that she was my first love. But I also am so mad at sometimes. We had talked about me going off to college and we knew that it probably would work about, so we decided that we would date other people. From my experience this really does work out. The first girl that I dated after her was a girl from my waiting job in New Braunfels. I decided that I should tell my exgirlfriend, whose name is Genie, about the girl. This was a very big mistake. Genie came to the restaurant where I worked and caused a big scene. But this isn't the only thing that makes me mad. Things are totally different now that we decided to see other people. We don't get along and we can't talk to each other. I think women need to just make up their mind. They all act like want this perfect gentlemen that does everything for them, but when the actually get that they don't know how to treat it. Usually the go to far and try to take advantage of it and then the guy starts to despise the girl. I don't really wish that things were back the way they were, I just wish that we could still get along. I really miss talking to her. She was a person that I could tell everything to and still feel comfortable about doing so. I am lucky though, because I have a sister that I am very close to. She also goes to UT and she has been a very big help with

> getting me settled in here in Austin. She only lives a couple of blocks away from me and she is there for me whenever I need anything, as I am for her. This is my freshman year and I am already dreaming that college would be over. It isn't that I don't enjoy Austin or College, it is just that I am tired of school. I wish that there could be a step in your life that you could just skip, but that is impossible. I would love to just be able to be settled in to a good paying job, but since that will never happen I am prepared to work now to enjoy the benefits later.

This particular instance is designated as negative within the category of "Openness." Nonetheless, our model has made an incorrect prediction by classifying it as positive in this specific class.

6.3.8.3.2 Extraversion

> Right now, I am sitting here sick to my stomach and the world feels so small. I am waiting for a phone call that is so important, and if I don't get it, I am going to feel like a really big loser. Yes, I did just get all the blessings I could ever ask for, so I am selfish to be wanting more, but its something I really really want. All I want is to make my parents proud and to give my family something they can brag about. I have spent my whole life wanting to achieve the best, and I get so sick when I let myself down. Rejection sucks. its so hot in here, and as all my friends call because they just got the call," I feel like a loser. I am proud of myself- but rejection is not something I handle well? What if the call does not come– will I cry, will I blame my inabilities on something else, how will I react? The anxiety I feel right now is extreme. On top of all that, I am homesick. I have a great life here in Austin, but since my family is a huge part of my life, I feel kind of left out being so far away. Everything back home seems to go on without me. my roommate here is annoying and the tv here is always on. she follows me around and sometimes I feel used because she really does not know people here. She is not in a sorority and so sometimes I feel as if she is angry at me for that. I am so anxious. my boyfriend is supportive too, but I wonder sometimes if he really has deep feelings for me. Yes, I know about his fear of commitment and all that crap, but we have been together for way too long for me not to feel totally secure with him. Oh, that stupid seventh heaven song. turn off the dang tv. All I want is peace and quiet without all the noise. Oh, and I have to worry about yesterday too. My sorority is awesome, but it makes me really uncomfortable to drink around some of them. Yes, I know. Its silly if we all drink together. But, sometimes I feel as if I have this

> image that I have to uphold. and that image reflects back onto all aspects of my life. my family, my faith, my school, my friends. How do I act? How do I dress? Who do I associate myself with? All of these things constantly flood my brain, and sometimes all I want to do is get far away from those thoughts. Do people love me for me? Do they love me for who I am here or the grades I make or the house I live in or the money my parents make? How do people view me? And that tv, always on. what I would give for that chatter to stop for 10 minutes. I can't even study with the noise. I am worried about this year. I need a job, I have bills to pay, I am in hard classes. how will I measure up? I love my life, I love my life. but I could seriously do without the stress. I am determined, and I already have accomplished so much this semester, but will it end? I want it to stay this way, but there is so much to lose. I am scared that I will lose it. How do I not lose it? I pray all the time, and I count my blessings. its hot in my apartment and it smells like paint. why did I choose to live in an apartment with a girl I don't like? What possessed me to do this? Did I feel independent and like a big girl? Now I feel young and naive, and way out of my league. oh, the insanity, but good things come to those who wait and I put all my trust into a higher being so things WILL work out.

This statement has been categorized as positive within the "Extraversion" class; nevertheless, our model has erred in its prognostication, misclassifying it as negative.

6.3.8.3.3 Conscientiousness

> ever since my boyfriend got this new job as a community assistant in an apartment complex, it doesn't seem like he has any time left over to spend with me. also, since he is a higher rank in rotc, he is even busier. so i question. what's going to happen to us? i ask him over and over again and he just gets upset. what am i supposed to think? every time this happens, we end up in an argument and threaten to break up which really hurts. i mean, he can't play with my emotions like that. it's not fair that he can have me waiting for him and giving up all my other plans in the hope that maybe this time, he'll come see me or make plans with never happens. it's not fair how he can just have me on the side when it's convenient to him. why is is that he seems like a totally different person now. not the same from the guy that i met more than a year ago. how can someone just change overnight? i am upset that when he does come and see me, it's is timed cause he says he's trying to squeeze me into his busy schedule. it make me feel like i am in prison and getting visitation rights or something. relationships shouldn't be like that. it

was never like that in the begining. but he says he's a different person now. he just called right now and hung up on me because i told him i couldn't talk cause i was doing this thing for the psychology class. he's mad. but what am i supposed to do? after all, the reason i am here, is to go to school and learn and stuff. if he expects me to understand everything he does why can't he understand that i need to do this thing. i feel like i'm gaining a little bit of weight and that bothers me a lot. yet, i'm too stubborn to get into a diet and too lazy to go excercise at the gym. i am sooooo stresed out. not just from the crap i have to put up with my boyfriend but also because of school work and the crap i have to put up to with work. work does not seem fun anymore. it was in the begining when i first started working there for more than a year ago. maybe because it was my very first job and i was getting paid more that i thought i would be. or maybe it was cause i'm new in town and was meeting lots of people then who are my age. but now, it seems like work is just a drag. maybe i'm jealous cause my boyfriend has this wonderful job or may be it's cause a lot of the people and managers that i started working with left to another state or for another occupation and just wanted to get away. i need the money that is why i am still working there. i applied at the hospital a couple of weeks ago but they haven't called me back or anything. then last week, i decided i wanted to volunteer at the children's hospital and when i called to inquire about it to see what i got to do, they told me that they were good. they were good? how can that be. they're a hospital. i thought they always needed help. and i was going to do some services for free. it's not like i was going to ask pay or anything. it was going to be free. my boyfriend's roomate's mom works there and the roomate had told me that he was going to ask his mom to give me a job and he did and she said that all i needed was to give her the hours that i can work. i mean, i can do that but it would be really awkward in my position because the mom is my boyfriends ex mom. i just didn't want to be in that position you know? and i really need to start working in the nursing field and get out of being a cashier at heb because that's my major, nursing. that's another thing i was worried about. what if i don't get accepted to nursing school next semester? then what am i going to do? maybe i can switch to pharmacy just like what my friend did. but i don't think it will be any easier or anything.

This statement is annotated as non-conscientiousness in the corpus. However, our model has predicted the personality trait of the author of this statement as having conscientiousness.

6.3.8.3.4 Agreeableness

I thought I would because I've visited with my friends so many times before, but now that I'm actually here it's finally true. I'm away from my parents, it's so great. I live with three great girls in my suite and we're so popular here. I've always been a socially outgoing person, but now I feel like it's going to work. there are always large numbers of people in our living room, bringing in food or beer to contribute to our refrigerator; everyone munches from it. and it's OK. the RA told us about this girl in another room who got so upset because her roommate ate her store bought cookies without asking; she called her mom and was so upset. I'm so glad its not like that here. we all contribute and all consume. But it's not like there's always noise and party's here. only when we all decide. if one person wants to read or study or sleep, we're really considerate. I hope that lasts, I'm pretty sure it will. At our building there are many foreign exchange students which is always a plus because, come on, who minds a foreign accent every once in a while. this guy from Belgium and this one from England are always watching TV in our room, which is another amusing thing: we don't have cable, or an antenna, or a VCR, so we only get FOX Channel 7. We sit around and watch whatever's on. in one way it's good because we don't have arguments over which channel to watch. maybe simplicity is the root of compromise. We had a floor meeting the other night here and they discussed some issues that had come up. it was so funny because almost all of them referred to our room's shananagans. This one guy came here from where he lives in a house to use the laundry (he's one of our friends- our referring to my roommate and I we've been friends since 2nd grad, long time, huh?) anyway, he dropped like half a box of laundry detergent on the stairwell and no one noticed for a week. the RA got mad and cleaned it up herself, but it was amusing because he doesn't even live here. another thing was the "stolen furniture" incident. we are given this loveseat-type couch in our suite's living room that can maybe seat 3 people if you're lucky. and in the lobby of the 3rd floor in front of the elevator there are 2 large couches that just block the pathway, no one ever sits in them, and they could probably seat 5 or 6. so when no one was around, my suitemate and I and 3 other people that happened to b in our room at the time helped us move our dinky little couch into the lobby which is down the hall and around a corner. we hauled the large couch down the way and we had to tilt it sideways and temporarily knock off some of the ceiling tiles just to make it in the doorway without banging down the door across from us. now

> we have a nice couch that is well used and the RA's are threatening to do a room check to find it. why? its going to more use. It's all kind of a double standard anyway. The head RA is always in our room hanging out and drinking our beer. he has a crush on me so he always brings us stuff and won't mention the couch to the others and lets us into the cafeteria at night. it's pretty funny, one night the night guard knocked on our door because someone had made a noise complaint. we opened the door and the guard stood in the threshold and the head RA stood behind the door quietly while we got reprimanded. it probably wouldn't have been in his best interests to b seen in there. He's only 20, but the building is changing management, so right now he's the head guy. its odd. I'm 18. finally. I could be in a management position at the pool I lifeguard at in the summers, next summer. it seems odd that I'm really an adult. when you're a kid u never think that you're ever going to get to the point where you decide when to come home and when to do this and what to do in this situation, type thing. its like the transition from high school to college really is that much of a change in that you're independent. it feels so good to finally b independent, financially, physically, emotionally. its wonderful responsibility. I am responsible for watching my budget, if I don't, no one will bail me out (well that's probably not true but you know). I guess I'm trying out freedom on borrowed wings, I can always have that security blanket if I want, but I don't want. I want to be independent. I am right now, I hope to stay that way.

This person's personality trait shouldn't be agreeable. However, our model has misclassified it.

6.3.8.3.5 Neuroticism

> Every day is a rollercoaster of emotions for me. I wake up in the morning with a knot in my stomach, fearing what challenges the day might bring. Even the simplest decisions can send me into a spiral of doubt and anxiety. It's as if a never-ending storm of worry and fear rages inside me. Social interactions are a minefield; I'm constantly second-guessing what I say and how others perceive me. I replay conversations in my head, dissecting every word for hidden meanings or signs of disapproval. Criticism, no matter how constructive, feels like a personal attack, and it takes me days to recover from it. I often find myself unable to let go of past mistakes, no matter how trivial. My mind races with 'what ifs' and 'should haves.' It's a daily struggle to keep my anxiety in check and maintain a semblance of normalcy, but most days, it feels like a battle that I'm losing.

This extended paragraph provides a more detailed and vivid description of a person's experience characterized by high levels of anxiety, constant self-doubt, and sensitivity to social interactions and criticism, all of which are indicative of the neuroticism personality trait. However, our model misclassifies it.

6.4 Conclusion

The task of identifying psychological traits is approached from two distinct directions in this study. The initial strategy employs a siamese architecture to train statement encoders, aiming to bring statements closer to their respective baseline statements in the vector space. Various implementations of siamese architectures, including Bi-LSTM and different Sentence-BERTs (SBERTs), were experimented with, revealing the superior performance of BERT-based models in this context. However, a limitation of BERT-based models is their token input constraint of 512 tokens. To process statements exceeding 512 tokens, a common occurrence in this task, the statements are truncated to 512 tokens.

To address this limitation, the next study explores the utilization of an hierarchical structure comprising tree-transformers and a graph attention network (GAT) to reframe the personality trait identification challenge as a multi-label classification problem. The tree-transformers, operating on a single sentence at a time, are employed to generate sentence embeddings in the initial layer of the hierarchy. Subsequently, GAT is applied as an overarching mechanism to derive statement embeddings from the generated sentence vector representations.

In the final study, an heterogeneous GAT is employed to refine word embeddings based on statement and sentence embeddings, thereby further enhancing the word embeddings with contextual information. Through this approach, the model achieves state-of-the-art performance across three benchmark corpora spanning two personality trait models: OCEAN and MBTI.

Chapter 7

Scientific Article Summarization

Summarization is a process that involves condensing a text, like an article or a book, into a shorter version that retains the main ideas and key points. Scientific document summarization presents a greater challenge compared to summarizing short text due to the length of text and the complexity and technical nature of scientific language. Understanding that this type of document is part of a body of writings that cite each other, we have introduced a corpus containing 10,000 research articles with their corresponding citing statements, and integrated a citation network in the models to provide background information. This chapter combines three of our publications for scientific document summarization (i) **"Generating Extractive and Abstractive Summaries in Parallel from Scientific Articles Incorporating Citing Statements"**, (ii) **"Enhancing Scientific Document Summarization with Research Community Perspective and Background Knowledge"**, and (iii) **"Investigating Semantic Similarity-Induced Parallel Training of Abstractive and Extractive Scientific Document Summarizers"**.

In the first two works, our proposed models generate in parallel the two types of summaries, extractive and abstractive. This parallel training mechanism allows the counterparts to attain a performance boost. All of these models utilize heterogeneous GAT to consider the inter-sentence relations and relations between sentences and the words. These word and sentence nodes are initialized by the 768-dimensional word tokens and sentence representations ([CLS] token) of the Longformer architecture.

In the third work we have introduced a loss function that considers the semantic distance between the generated and the reference summaries so that the generated summaries become semantically more similar to the reference summary. To assess the impact of the parallel training approach and loss function, in this work we have experimented with four state-of-the-art (SOTA) extractive and four SOTA abstractive summarizers. The experimental results show that with this proposed training approach the SOTA models have gained significant performance boosts.

7.1 Generating Extractive and Abstractive Summaries in Parallel from Scientific Articles Incorporating Citing Statements

This section is based on the paper titled "Generating Extractive and Abstractive Summaries in Parallel from Scientific Articles Incorporating Citing Statements" co-authored with Robert E. Mercer that appeared in *The 4th New Frontiers in Summarization Workshop (NewSumm 2023)* [206].

Summarization of scientific articles often overlooks insights from citing papers, focusing solely on the document's content. To incorporate citation contexts, we develop a model to summarize a scientific document using the information in the source and citing documents. It concurrently generates abstractive and extractive summaries, each enhancing the other. The extractive summarizer utilizes a blend of heterogeneous graph-based neural networks and graph attention networks, while the abstractive summarizer employs an autoregressive decoder. These modules exchange control signals through the loss function, ensuring the creation of high-quality summaries in both styles.

7.1.1 Introduction

Text summarization automates condensing documents while preserving key information. Most neural summarization models, like those by Nallapati et al. [155], Zhong et al. [272], etc. are designed for shorter texts, e.g., the CNN/Daily Mail dataset [83]. However, applying these models to longer documents, such as scientific research papers, remains limited. In scientific document summarization, it is common to focus solely on abstracts, introductions, and conclusions, as demonstrated in Yasunaga et al. [251]'s work.

Summarizing scientific publications presents unique challenges due to their length, complex concepts, technical jargon, structured organization, and citations. These complexities make it a more daunting task compared to summarizing other types of documents. Additionally, the long-term impact of a scientific article may not be fully evident when it is first published, as its significance can evolve over time. While an abstract provides an initial overview from the authors' perspective, it may not capture the full extent of the paper's influence on the research community and its evolving impact [251]. As an example, we can consider the abstract from Bergsma [24]:

> We present an approach to pronoun resolution based on syntactic paths. ... we learn the likelihood of coreference between a pronoun and a candidate noun based

> on the path in the parse tree between the two entities. ... Highly coreferent paths also allow mining of precise probabilistic gender/number information. We combine statistical knowledge with well known features in a Support Vector Machine pronoun resolution classifier. Significant gains in performance are observed on several datasets.

This abstract gives insight into the methods the authors used. But the citations emphasize the corpus it presents. For example:

> We use the approach of Bergsma and Lin (2005), both because it achieves state-ofthe-art gender classification performance, and because a database of the obtained noun genders is available online. [25]

> For the gender task that we study in our experiments, we acquire class instances by filtering the dataset of nouns and their genders created by Bergsma and Lin (2005). [26]

Jaidka et al. [93, 94] have identified this missing aspect in scientific document summarization and addressed it by introducing a shared task. This task aims to create summaries that take into account not only the information in the body of the documents but also the research community's overview of the documents over time. The work described here continues in this direction.

With the advancement of neural networks, there have been a few prominent research works in recent years for generating extractive [251] and abstractive [257, 264] summaries from scientific documents [50, 262]. Extractive summarization recognizes key sentences from the source document as the summary but lack the flow of information, whereas the abstractive summarization technique generates new phrases using language models while preserving the semantics of the input document but may miss some important aspects of the text. This is a motivation for designing a model to generate both summaries in parallel and help the counterpart to achieve a performance boost with additional guidance.

A key step in extracting brief synopsis sentences from a manuscript is to map the cross-sentence correlations. A lot of recent prominent works [154] have tried to do so using recurrent neural networks (RNNs). However, because of using RNNs, these models fail to capture long-distance sentence-level dependencies. Another approach to preserve sentence-level dependencies from long documents is using graph-based neural networks. A few recent works (e.g., [51, 252]) have utilized discourse information in the article along with inter-sentence correlations for constructing graphs and summarizing document. Another approach is to construct a sentence-level fully connected graph. Zhong et al. [272] and Liu et al. [126] used

transformer [223] encoders to determine how sentences interact with each other. Wang et al. [227] introduced an heterogeneous graph neural network for extractive summarization which used additional semantic units (words) as intermediate nodes to construct relationships between sentences.

Abstractive summarizers focus heavily on form, with the goal of producing a generalized summary, which tends to necessitate complex language-generating models. These models are typically based on sequence-to-sequence (seq2seq) architectures, in which a source document is seen as one sequence whereas its summary as another. The majority of previous research on neural abstractive summarization depended on large-scale, high-quality datasets of supervised document-summarization pairings [197]. Recently, state-of-the-art solutions on abstractive summarization are built upon the transformer [223] and BERT [56] models. These attention-based abstractive models are being used in different fields like clinical note summarization [102], scientific document summarization [262], and lay-abstract generation [257].

In this paper, addressing the above-mentioned issues, we have built a standalone summarization model which can generate both extractive and abstractive summaries from scientific documents incorporating the citation network. Analyzing the citation network, citing statements from the citing articles are accumulated with the original text document to incorporate the research community's observation on that particular cited manuscript. These summaries are the abstracts of the original papers with additional information reflecting the research community's view. After that, we run the LongFormer [22] encoder to generate sentence and word representations and train extractive and abstractive summarizers together. For the extractive summarizer, an heterogeneous graph neural network [227] is used as it has the ability to preserve sentence-level dependencies utilizing additional semantic units as intermediate nodes in the graph representation. Abstractive summaries are generated by the autoregressive decoder. The loss function is defined in such a way that both summarizers can achieve better ROUGE and METEOR scores. Furthermore, we have developed a corpus containing 10K research articles along with their corresponding citation statements and is a subset of the Semantic Scholar Network (SSN) corpus. The citation statements are collected utilizing the citation graph used in the SSN corpus. In short, the contributions of this work are:

- We have built a stand-alone summarizer model which can produce both extractive and abstractive summaries and each counterpart helps the other to generate better summaries.
- The summarizer model can work with long scientific text articles
- This model considers research communities' observations while generating the summaries

- We have proposed a new corpus containing 10K research articles along with the corresponding citing statements to incorporate the research communities' view.

7.1.2 Related Work

Text summarization aims to distill a document's essence efficiently. Recent NLP research has yielded effective neural summarization models, particularly those using transformer and BERT-based architectures. Work summarizing lengthy scientific documents often focuses on specific sections rather than the entire text [257] or citation statements [10].

7.1.2.1 Extractive Text Summarization

Extractive text summarization models classify sentences in a document using labels that indicate whether or not a sentence ought to be included in the summary. Originally, these models were designed based on the encoder-decoder architecture using RNNs [154]. Since transformer and BERT-based models provide a more enriched sentence encoding, they have become the foundation for the majority of extractive summarizer models in recent years. Liu et al. [126] fine-tuned BERT with stacked layers of transformer to obtain the sentence vectors and then used a sigmoid classifier for identifying the sentences that would be included in the summary. Zhang et al. [264] fine-tuned an hierarchical transformer (HIBERT) for the extractive summarization task. Another prominent approach for extractive summarization is using graph representations which can preserve sentence-level correlations. Later, the graph convolutional network (GCN) [232] has been espoused for building different inter-sentence correlation graphs [252] for this task. Wang et al. [227] built an heterogeneous graph neural network for extractive summarization (HeterSumGraph) which takes into account additional semantic units at the word level for building the sentence-level correlation graph.

7.1.2.2 Abstractive Text Summarization

Abstractive text summarization models, unlike the extractive summarizers which work like classifiers, are intended to generate summaries comprising new sentences which may or may not be present in the body of the document. These models are mostly based on the encoder-decoder architecture of the sequence-to-sequence models and language models like BART [115], BigBird [258], and T5 [178]. Aksenov et al. [8] applied BERT-windowing to overcome the length limitation of the BERT model and summarize long documents. Gidiotis et al. [73] trained the summarizer model to generate separate abstractive summaries for small parts of the document. Pilault et al. [170] combined both the extractive and abstractive summarization using a transformer language model and built an hybrid summarizer model. Yu et al. [257]

fine-tuned pre-trained BERT as the abstractive summarizer for generating a lay summary from the document.

7.1.2.3 Scientific Article Summarization

Existing scientific article summarizers, in most cases, are extractive models designed on the idea of sentence selection [51]. Coha et al. [50] developed the first abstractive summarizer for long scientific articles using an hierarchical encoder and discourse-aware attentive decoder. Mishra et al. [148] applied citation contextualization to extract unique relevant sentences from the document and final summaries are generated using a multi-objective clustering approach. Gupta et al. [81] applied BERT and graph-based approaches for biomedical document summarization. Li et al. [120] fine tuned T5 for generating summaries from long scientific documents and implemented an extractive summarizer using GCN. Yasunaga et al. [251] built a corpus (ScisummNet) that includes a citation network for scientific document summarization and extracted the summary-candidate sentences using a GCN. An et al. [10] introduced a large corpus (SSN) with 141K research papers connected with a citation graph. They also proposed a graph-based summarization model (CGSUM) for extractive document summarization. This model can draw information from both the source and the citing texts.

7.1.3 Methodology

This section defines the problem of scientific document summarization using a citation graph. Then, the two benchmark datasets used for the scientific article summarization experiments are discussed along with the pre-processing procedures. Finally, the proposed deep learning model is explained.

7.1.3.1 Problem Formulation: Summarization Using Citation Graph

Scientific articles possess distinctive attributes, including citation linkages, that establish profound connections between their contents. These studies may also yield unforeseen impacts and evolve in importance as research progresses. In such cases, ideal summaries should encompass both the authors' key points and the perspectives of the scientific community, as reflected in citations [251]. To serve this intent we have utilized two resources: the citation graph provided in the Semantic Scholar Network (SSN) corpus [10], and the ScisummNet/CL-SciSumm-2020 (CL-SciSumm-2020) corpus [41, 251] which supplies documents and their corresponding citing statements.

7.1.3.2 Description of the Datasets

As this work is focused on generating summaries from scientific articles that incorporate the research community's views, we have considered two benchmark datasets: ScisummNet/CL-SciSumm [41, 251], and Semantic Scholar Network (SSN) [10] for the experiments done here. To the best of our knowledge, these are the only datasets for the summarization task that also provide citation information. The ScisummNet corpus consists of abstracts of the 1000 most cited research articles from the ACL Anthology Network [177] along with 15 citing statements per article. The gold standard summaries for these 1000 documents are manually summarized by domain experts. The CL-SciSumm-2020 corpus [41] extends the ScisummNet corpus with 40 extra documents and human-generated summaries thereby providing 1040 documents, citation sentences, and summaries. For testing, we have used the test set comprising 200 scientific articles from the CL-SciSumm-2020 corpus. The other benchmark dataset used for this task is the SSN corpus. It includes 140,799 research articles culled from the Semantic Scholar Open Research Corpus (S20RC) [131] together with a large citation graph. This citation graph has each article as a node and 660,908 edges indicating the citations. This corpus covers research articles from three domains: physics, mathematics and computer science.

The primary objective of this study is to develop a deep learning model capable of generating summaries for lengthy scientific documents while incorporating insights from other researchers citing the document. While the ScisummNet/CL-SciSumm dataset provides citation statements, the SSN corpus lacks this information. Originally, the SSN corpus consisted of documents and their references, but for our purpose of including citing statements, modifications were necessary. We leveraged the citation graph to identify citing papers and manually extracted the statements referring to the cited articles. Given the substantial size of the SSN corpus, containing nearly 141K articles, we randomly selected 10K papers for summarization. These papers have body lengths ranging from 1000 to 3500 words (with background/related work sections removed), aligning with the capacity of the LongFormer model (as described in Section 7.1.5), which can handle a maximum of 4096 tokens at a time. The dataset was divided into training (8000), validation (1000), and testing (1000) articles to facilitate model development and evaluation.

Citations can convey positive, neutral, or negative intentions. To capture this diversity, we systematically categorized citing statements into these three classes after gathering them from citing articles. In cases where a paper had limited negative citations, we balanced the selection by including more neutral and positive citation statements. To classify these citation statements, we have employed RoBERTa trained on Athar [17] following the approach used by Kundu [113].

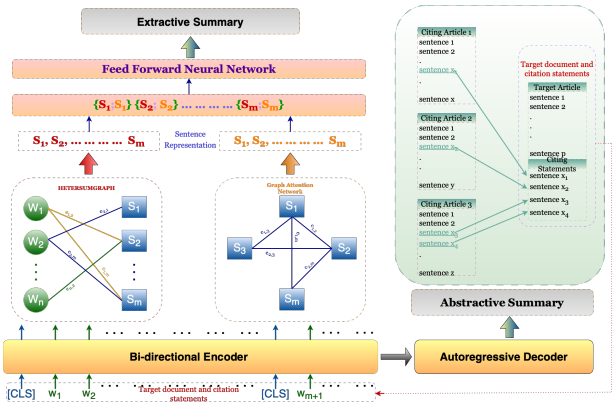

Figure 7.1: System architecture of the proposed model

In the SSN corpus, the summaries are limited to the authors' perspectives as they consist of the paper abstracts. To create more comprehensive summaries, we employed a two-step approach. First, we used a fine-tuned T5 model [178], trained on the CL-SciSumm-2020 corpus, to generate five summaries per document by inputting both the abstracts and corresponding citation statements. Then, we have employed a pre-trained RoBERTa architecture to obtain five vector representations for these summaries. The most similar summary to the reference summary, determined by cosine similarity, was selected as our T5-Generated Summary.

7.1.4 Model Overview

The investigated summarization model has two units: an extractive and an abstractive summarizer. The overall architecture of the model is portrayed in Figure 7.1. This section discusses the architecture and working principle of these two units.

While designing the extractive summarizer, we have considered two issues: how the sentences are connected to each other and how semantic units like words affect the sentence level correlations. To fulfill these purposes, we have utilized two different graph-based neural networks: an heterogeneous graph neural network (HeterSumGraph) [227] and a graph attention network (GAT) [224].

For any graph $G = \{V, E\}$, V denotes the nodes and E, the edges between them. HeterSumGraph defines $V = V_w \cup V_s$, V_w is the set of unique words and V_s is the set of sentences in the document. For a document with n unique words and m sentences, E is the edge weight matrix, where $e_{i,j}$ represents word i in sentence j, ($i \in \{1 : n\}, j \in \{1 : m\}$) [227]. The nodes that represent the sentences are initialized with LongFormer [CLS] tokens. Because LogFormer generates a contextualized word embedding for each occurrence of the word in the document, all of the word embeddings for a word are averaged to initialize that particular word-representing node in the graph. The edges between the words and sentences are initialized with

the corresponding TF-IDF values.

After the graph G is constructed, a graph attention network (GAT) is used to update the node feature values. Considering $h_i \in \mathbb{R}^{d_h}$ where $i \in \{1 : (n + m)\}$ as the hidden states of the word and sentence nodes, the GAT layer is designed as:

$$\mathcal{T}_{i,j} = LeakyReLU(\omega_a[\omega_q h_i; \omega_k h_j; e_{i,j}]) \tag{7.1}$$

$$\alpha_{i,j} = \frac{\exp(\mathcal{T}_{i,j})}{\sum_{l \in \mathcal{N}_i} \exp(\mathcal{T}_{i,l})} \tag{7.2}$$

$$u_i = \sigma(\sum_{j \in \mathcal{N}_i} \alpha_{i,j} \omega_v h_j) \tag{7.3}$$

where ω_a, ω_q, ω_k and ω_v are learnable weight matrices. $\mathcal{N}_i$ denotes the list of the neighbor nodes. The attention value between h_i and h_j is denoted by $\alpha_{i,j}$. The GAT with multi-head attention (considering $\mathcal{K}$ attention heads) is designed as:

$$u_i = \|_{k=1}^{\mathcal{K}} \sigma(\sum_{j \in \mathcal{N}_i} \alpha_{i,j}^k \omega^k h_i) \tag{7.4}$$

To prevent the gradient from vanishing, HeterSumGraph incorporates a residual connection and the final hidden state representation becomes:

$$h_i = u_i + h_i \tag{7.5}$$

Through the aforementioned GAT and position-wise feed-forward network (FFN) layer comprising two linear transformations [227], the sentence nodes are updated with their adjacent word nodes:

$$\mathcal{U}_{w \to s}^1 = GAT(\mathcal{H}_s^0, \mathcal{H}_w^0, \mathcal{H}_w^0) \tag{7.6}$$

$$\mathcal{H}_s^1 = FFN(\mathcal{U}_{w \to s}^1 + \mathcal{H}_s^0) \tag{7.7}$$

where $\mathcal{U}_{w \to s}^1 \in \mathbb{R}^{n \times d_h}$ $\mathcal{H}_w^1 = \mathcal{H}_w^0 = V_w$, and $\mathcal{H}_s^0 = V_s$. In Eq. 7.6, $\mathcal{H}_s^0$ is employed as the attention query and for both the attention key and value $\mathcal{H}_w^0$ is used. Then, the revised sentence nodes are used to generate new representations for the individual word nodes and continue to refine the revised sentence nodes in an iterative fashion. At each iteration, sentence-to-word and word-to-sentence updates continue to be processed. The process can be depicted as follows for the

t-th iteration:

$$\mathcal{U}_{s\rightarrow w}^{t+1} = GAT(\mathcal{H}_w^t, \mathcal{H}_s^t, \mathcal{H}_s^t) \tag{7.8}$$

$$\mathcal{H}_w^{t+1} = FFN(\mathcal{U}_{s\rightarrow w}^{t+1} + \mathcal{H}_w^t) \tag{7.9}$$

$$\mathcal{U}_{w\rightarrow s}^{t+1} = GAT(\mathcal{H}_s^t, \mathcal{H}_w^{t+1}, \mathcal{H}_w^{t+1}) \tag{7.10}$$

$$\mathcal{H}_s^{t+1} = FFN(\mathcal{U}_{w\rightarrow s}^{t+1} + \mathcal{H}_s^t) \tag{7.11}$$

Once the model training is done, the sentence nodes' representations are used as the sentence vector representations.

For direct sentence-level interactions, we have also used a graph attention neural network (GAT). Here, for the graph $G = \{V, E\}$, $V = V_s$ where V_s is the set of all the sentences in the document. The edge weight matrix E preserves the semantic similarity values between sentences. The nodes are initialized in the same manner as the sentence nodes in HeterSumGraph. For initializing the edges between nodes, at first we have acquired the vector representations of the sentences using pre-trained LongFormer and then computed the cosine similarity between the sentences. The edges are initialized with the corresponding similarity values between sentences. However, as scientific documents come with many sentences, working with a fully connected graph is not computationally cost effective. To reduce the burden of computational overhead, we have dropped the edge connections between nodes whose cosine similarity values are below a certain cut-off value. Throughout the conducted experiments, we have found that if we set the cut-off value below 0.3, the performance of the summarizer model remains the same.

Considering node features $h = \{h_1, h_n, ..., h_m\}$ as the input, GAT applies a self attention on the nodes and computes the attention coefficients as follows:

$$\mathcal{T}_{i,j} = a(\omega h_i, \omega h_j) \tag{7.12}$$

where a is a single-layer feed forward neural network with the *LeakyReLU* activation function, and ω is a learnable parameter. This attention coefficient shows node j's importance on node i and it is computed only for the corresponding one-hop neighbour nodes ($j \in \mathcal{N}_i$). This attention coefficient value is normalized to compute the attention values as follows:

$$\alpha_{i,j} = \frac{\exp(\mathcal{T}_{i,j})}{\sum_{l\in\mathcal{N}_i} \exp(\mathcal{T}_{i,l})} \tag{7.13}$$

The multi-head attention is computed in the same way it has been done for HeterSumGraph (Eq. 7.4).

Once the sentence representations from both the HeterSumGraph and GAT are computed, they are concatenated and fed to the feed-forward neural network layer. This is a two-layer position-wise feed-forward layer [227] for labeling the sentences with 1 or 0; 1 indicates that particular sentence is included in the extractive summary.

The abstractive summary is generated by the LongFormer decoder. To train the summarizer units in parallel, the training mechanism in Yu et al. [257] is used. The overall loss L of the model is:

$$L = L_{ext} + L_{abs} \tag{7.14}$$

where L_{ext} and L_{abs} represent the cross-entropy losses of the extractive and abstractive summarizers, respectively.

7.1.5 Experimental Results and Analysis

This section gives a brief description of the model parameters used in the experiments as well as the results achieved on CL-SciSumm-2020 and the customized SSN datasets.

7.1.5.1 Model Parameters and Training Details

We have trained our model on a 48GB NVIDIA RTX A6000 GPU. The batch size has been set to 1 as the length of input documents plus the citation statements is large. Since all the experiments are done on a small batch-size, we have followed the training procedure of Sefid et al. [198] and accumulated gradients for 10 steps and updated the parameters. The NOAM scheduler has been utilized to adjust the learning rate and gradients are clipped so that exploding gradients during training can be prevented. The model has been trained for 20,000 epochs. The extractive summarizer is initialized with the LongFormer embeddings. Following that, the LongFormer encoder-decoder architecture for the abstractive summarizer and the extractive summarizer units' forward passes are trained separately. Once both of the forward passes are done for each iteration and the individual losses are calculated, the model's overall loss is calculated. If either of the two unit's validation loss continues to go down for 5 epochs, the parameter settings for that particular unit are saved and that unit's training is postponed for the next 10 epochs. The number of attention-heads for multi-head attention has been set to 8. The stop words and punctuation have been filtered out when pre-processing the word nodes in the graph. Following Wang et al. [227], 10% of the words in the vocabulary having low TF-IDF values have been further filtered out. The word and sentence nodes have been initialized with 768-dimensional vectors. And the sentence representations from both the HeterSumGraph and

Table 7.1: Results on the modified SSN corpus. The results consider both the abstracts and the T5-generated summaries incorporating citation statements as the reference summaries. The best results are boldfaced.

Models	On Abstracts as Summaries				On T5-Generated Summaries			
	R-1	R-2	R-L	METEOR	R-1	R-2	R-L	METEOR
Extractive								
BERTSumExt	42.92	14.19	39.01	33.09	43.11	14.21	39.12	33.07
HeterSumGraph	44.27	14.52	39.73	33.18	44.30	14.53	39.74	33.18
GRETEL	**45.22**	**15.19**	**40.23**	**36.87**	**45.23**	15.19	**40.24**	**36.88**
Proposed Model (Extractive)	45.19	15.18	40.21	36.83	45.19	**15.21**	40.23	36.85
Abstractive								
PTGen+Cov	41.66	13.08	36.95	32.44	41.60	13.10	36.72	32.40
BERTSumAbs	42.06	14.52	38.17	32.49	42.04	14.56	38.17	32.49
BERT+CopyTransformer	42.43	15.01	39.03	32.88	42.44	15.05	39.04	32.91
Proposed Model (Abstractive)	**44.82**	**15.19**	**39.31**	**36.50**	**44.83**	**15.19**	**39.30**	**36.51**

GAT are 512-dimensional vectors. So, the final sentence vectors after the concatenation step are 1024-dimensional vectors. The Feed Forward Network hidden layer size is 512.

7.1.5.2 Performance Analysis of the Model

We have performed experiments on two datasets: modified SSN and CL-SciSumm-2020. The results achieved by our models are reported as overlapping unigrams, bigrams, and the longest common sequence between the generated summaries and the reference summaries by means of R-1, R-2, and R-L metrics; and semantic compatibility between the reference and generated summaries by means of METEOR metric, respectively, for the modified SSN corpus. R-1, and R-2 show the informativeness, and R-L shows the fluency of the generated summary. The metrics used for analyzing the model performance on CL-SciSumm-2020 are R-2 and R-SU4, which indicate the proportion of bigram overlap and unigram plus skipgram of 4 tokens overlap, respectively, between the reference and generated summaries. The performance here is also analyzed with the METEOR metric. As the Bi-directional encoder and autoregressive decoder we have also experimented with BigBird. However, the better performance was found with LongFormer. That is why in the final model, we have used LongFormer in all the cases for initial encoding and generating abstractive summaries.

7.1.5.2.1 Results: Modified SSN Corpus

To compare the performance of our model with the existing extractive models, we train and test the following extractive summarizer models on our modified corpus: (1) BERTSumEXT [126]: a BERT-based model; (2) HeterSumGraph [227]: a heterogeneous graph-based approach that considers the cross-sentence correlations using additional semantic units; and (3) GRETEL:

fuses semantic information from the document context and gold summary using a hierarchical transformer encoder and graph contrastive learning. For the abstractive summarization baseline, we have experimented with: (1) PTGen+Cov [197]: based on a hybrid pointer generator network to copy words from the source text, (2) BERTSumAbs [126]: a BERT-based model; and (3) BERT+CopyTransformer [8]: applies BERT-windowing for processing data longer than the BERT window.

The performance of the existing models and our proposed models are shown in Table 7.1. As reference summaries, we have considered both the paper abstracts as well as the summaries we have generated from the abstracts plus the citing statements using T5.

Although BERTSumExt and BERTSumAbs perform very well with short documents, their performance metrics are not at that level when summarizing scientific documents. The main reason for this is their limitation to working with a maximum 512 input tokens, but scientific documents are much longer. For this, they have applied the greedy algorithm introduced by Nallapati et al. [155]. HeterSumGraph considers direct relationships between words and sentences on texts with a 50-sentence maximum, whereas our proposed model considers direct cross-sentence correlations, as well, and can deal with longer text spans (up to 3500 words). These additional features, together with LongFormer's enriched word and sentence features, gives our model a performance boost, but our model requires more computational time and resources. Our model performs better by a good margin compared to the other models apart from GRETEL. Our extractive summarizer shows slightly lower performance compared to GRETEL which is a more complex model. Still, because of the parallel training approach, our model has achieved comparable results. Our abstractive summarizer model outperforms the other experimental abstractive summarizers by large margins: PTGen+Cov by 2.36, BertSumAbs by 1.14, and BERT+CopyTransformer by 0.28 R-L scores. The METEOR scores achieved by our model are 36.83 and 36.50 for extractive and abstractive summaries, respectively, when tested over the T5-generated summaries. In the experiment with the abstracts as summaries, the METEOR scores are 36.51 and 36.85 for the abstractive and extractive summaries, respectively. Looking at the METEOR scores achieved by the other models (see Table 7.1), it is clearly visible that both the extractive and abstractive summarizer units of our model have outperformed them by at least 3. This observation indicates that the summaries generated by our proposed model are more semantically similar to the reference summaries. To see the importance of the individual units, please check the ablation study in the appendix.

7.1.5.2.2 Results: CL-SciSumm-2020 Corpus

For analyzing our proposed model's performances on CL-SciSumm-2020 Corpus, we have used R-2 and R-SU4 F-1 scores (as the other comparable models are reported with these met-

Table 7.2: Model performance analysis on two CL-SciSumm-2020 summary categories. All values are F-1 scores.

Models	Abstracts as Summaries			Human-created Summaries		
	R-2	R-SU4	METEOR	R-2	R-SU4	METEOR
Jaccard-focused GCN	0.19931	0.09956	-	0.2042	0.14162	-
Clustering	0.1959	0.0962	-	0.1749	0.1169	-
MMR2	0.15067	0.07851	-	0.15073	0.10237	-
LSTM+BabelNet	0.329	0.172	-	0.241	0.171	-
Proposed Model						
Extractive Summarizer	0.43	0.266	31.12	0.42	0.249	30.18
Abstractive Summarizer	0.43	0.250	30.98	0.41	0.234	30.06

rics) We have experimented to generate abstract and human summaries. As benchmarks, we have selected the research works submitted to CL-SciSumm-2019/2020: (1) Jaccard-focused GCN [220]: an extractive summarizer utilizing cross-sentence graph and graph attention networks, (2) Clustering [147]: based on different clustering algorithms followed by sentence-scoring functions, (3) MMR2 [180]: based on the maximal marginal relevance technique, and (4) LSTM+BabelNet [46]: BabelNet vectors were used to train the LSTM. The CL-SciSumm task provides a performance metric evaluation script which is used to calculate the R-2 and R-SU4 values for the model-generated summaries against the test set.

Results on CL-SciSumm-2020 are reported in Table 7.2. Looking at the results, it is clear that our model outperforms the other existing extractive models on every measure. The R-2 and R-SU4 achieved for both of our model-generated extractive and abstractive summaries are very high compared to the other existing extractive models. And this is the case for both the original abstracts and the human-created summaries as reference summaries. For the human-created reference summaries, our extractive and abstractive summarizers have achieved 0.078 and 0.063 R-SU4 F-1 score gains, respectively, compared to the LSTM+BabelNet model, which comes with the best result among the other considered models. While considering the abstracts of the papers as reference summaries, these gains are 0.094 and 0.078, respectively. For the abstractive summaries, the METEOR score achieved by our model is 30.18 whereas for the extractive summaries, it has achieved a 30.06 METEOR score on the human-generated summaries. Over the abstracts of the papers, these scores are 31.12 and 30.98, respectively.

7.1.6 Conclusion and Future Work

In this paper, we have introduced a summarizer model considering two intentions: first, summarize scientific documents incorporating citation contexts, and second, build a summarizer model which can generate both extractive and abstractive summaries by means of parallel train-

ing so that both counterparts can gain a performance boost. For this, we have utilized both the sentence-sentence and sentence-word correlations. Furthermore, we have constructed a corpus comprising 10K scientific articles with their corresponding citation statements for the summarization task. The experimental results show that our model performs well compared to other well-known methods. Though this work considers the research community's observations (citing statements), it doesn't consider the background information (references presented in the target article). In our future work, we are planning to use both sides of the citation graph (references as the background knowledge and the citing statements as the research community's views) while summarizing a scientific article.

7.1.7 Limitations

Our experiments are limited to summarize long scientific texts only. We have not conducted any experiments with short target texts, consequently we are not sure how well the model may perform while summarizing short texts. We are also unsure how well this model may perform for extreme summary generation like TLDR [36]. Moreover, we have trained both the extractive and abstractive summarizer units for a large number of epochs. Though to prevent any unit from being over-fitted we have checked the curve of validation loss after every 5 epochs. This is very computationally expensive and demands a longer period of time for model training. Furthermore, no tests have been performed to see how the abstractive summarizer unit suffers from hallucination.

7.1.8 Appendix

7.1.8.1 Ablation Study

Table 7.3: Ablation Study: Rows labeled with † indicate the extractive summaries and rows labeled with ∗ indicate abstractive summaries.

Discarded Unit	**On T5-Generated Summaries**			
	R-1	**R-2**	**R-L**	**METEOR**
GAT†	44.86	14.9	39.96	36.52
HeterSumGraph†	44.78	14.81	39.84.	36.49
Extractive Summarizer∗	43.01	15.02	38.99	35.92
Abstractive Summarizer†	44.91	14.95	39.96	36.50

To portray a better grasp of each component's contribution in our suggested model, we have experimented with different units of our model separately and the results are reported in Table

7.3. All of these experiments are performed on the T5-generated corpus which combines the abstract of the paper along with the citation statements.

In our first experiment, we have discarded the GAT unit which works with cross-sentence relationships and kept only the HeterSumGraph for extractive summary generation. This time the performances of the model are lower than the reported results in Tables 7.1 (R-1: 44.86, R-2: 14.91, R-L: 39.96, and METEOR: 36.52) for our generated extractive summaries. Still, these results are higher compared to the original HeterSumGraph model. It shows, using the LongFormer encoder in the beginning and using the collective loss function for both the abstractive and extractive summarizer units play a significant role in the performance boost. And it also indicates that taking direct cross-sentence correlations into consideration provides some additional features to enrich the model which helps the model's performance to improve.

In the second experiment, we have discarded the HeterSumGraph unit and used only GAT in the extractive summarization unit. This time the performance metrics for extractive summaries are R-1: 44.78, R-2: 14.81, R-L: 39.84,and METEOR: 36.4. These values are comparably lower than we gained in the last experiment. The reason behind this incident is, though no direct cross-sentence relationships are present, HeterSumGraph, by 2-hop distance, considers the correlations between sentences.

The third experiment discards the extractive summarizer unit. The LongFormer abstractive summarizer unit achieves very poor R-1: 43.01, R-2: 15.02, R-L: 38.99, and METEOR: 35.92 scores compared to the proposed model. This poor performance demonstrates the importance of the information that the extractive summarizer provides the abstractive summarizer through the combined loss function.

Finally, we have discarded the abstract summarizer unit and used the combination of HeterSumGraph and GAT for extractive summary generation. During this experiment, the achieved R-1, R-2 R-L, and METEOR scores are 44.91, 14.95, 39.96, and 36.50, respectively, which are more than the cases for the three above-mentioned ablation experiments. It indicates the significance of training the abstractive summarization unit in parallel as well as using the cross-sentence and semantic unit-sentence correlations at the same time.

7.1.8.2 Validity Check of the Proposed Corpus

To ascertain the corpus's quality, a rigorous analysis was conducted on a statistically significant subset of the dataset, with a confidence level of 95% and a margin of error of 3%, aided by three human annotators. Within the vast pool of 10,000 summarization samples, a random selection of 400 was subject to annotation for this statistical inquiry.

Each annotator was tasked with evaluating whether the summaries generated by the T5 model effectively encapsulated the same information as the combination of the abstract and the

citing statements. The first annotator affirmed that 374 samples achieved this concurrence, the second annotator concurred with 368, and the third annotator with 371.

When comparing the assessments of the first and second annotators, it was determined they agreed that 368 samples were appropriately summarized, while 16 were not, resulting in a substantial Cohen's κ of 0.89. In the comparison between the second and third annotators, a significant concurrence emerged for 396 samples, where 368 were accurately summarized, and 28 were not, yielding κ value of 0.93. Similarly, when examining the assessments of the first and third annotators, agreement was established for 398 summaries, with 370 being correctly summarized and 27 not, resulting in κ of 0.94.

7.2 Enhancing Scientific Document Summarization with Research Community Perspective and Background Knowledge

This section is based on the paper titled "Enhancing Scientific Document Summarization with Research Community Perspective and Background Knowledge" co-authored with Robert E. Mercer. Currently, this paper is under review for conference paper publication.

Scientific paper summarization has been the focus of much recent research. Unlike previous research which summarizes only the paper in question, or which summarizes the paper and the papers that it references, or which summarizes the paper and the citing sentences from the papers that cite it, this work puts all three of these summarization techniques together. To accomplish this, we have, by utilizing the citation network, introduced a corpus for scientific document summarization that provides information about the document being summarized, the papers referenced by it, as well as the papers that have cited it. The proposed summarizer model utilizes the referenced articles as background information and citing articles to capture the impact of the scientific document on the research community. Another aspect of the proposed model is it's ability to generate both the extractive and abstractive summaries in parallel. The parallel training helps the counterparts to improve their individual performance. Results have shown that the summaries are of high quality when considering the standard metrics.

7.2.1 Introduction

Text summarization represents an intricate procedure that entails the automatic condensation of a document, all the while preserving a succinct and coherent rendition of its content. In contrast to the widespread utilization of neural text summarization systems for brief texts

[155, 272], their application to longer documents, such as scholarly research publications, has not been markedly prevalent. In the context of summarizing scientific manuscripts, the prevailing method typically involves the selective extraction of content solely from the abstract, introduction, and conclusion segments within the target articles [251].

Scientific publications are characterized by their length, complex concepts, and domain-specific knowledge. They follow a structured format with sections and citations, serving to explain the subject matter to knowledgeable readers while meeting publisher-imposed page limits. Furnishing summarization models with this context is crucial for enhancing summary quality [10]. Consequently, summarizing scientific articles presents a more daunting task than other document types.

Moreover, there exists a latent dimension to the impact of any given scientific article at the point of its initial publication, which may become apparent only in subsequent studies by other researchers. While a paper's abstract provides a valuable snapshot of the content as envisioned by the authors, it may fall short in capturing the genuine influence that the paper might wield within its domain over time. This influence has the potential to evolve and assume different dimensions as it reverberates throughout the research community [251]. For instance, we can examine the abstract presented by Bergsma and Lin [25]:

> We present an approach to pronoun resolution based on syntactic paths. Through a simple bootstrapping procedure, we learn the likelihood of coreference between a pronoun and a candidate noun based on the path in the parse tree between the two entities. This path information enables us to handle previously challenging resolution instances, and also robustly addresses traditional syntactic coreference constraints. Highly coreferent paths also allow mining of precise probabilistic gender/number information. We combine statistical knowledge with well known features in a Support Vector Machine pronoun resolution classifier. Significant gains in performance are observed on several datasets.

This abstract provides a glimpse into the methodologies employed by the authors, whereas the citations underscore the significance of the corpus it presents. For instance:

> For the gender task that we study in our experiments, we acquire class instances by filtering the dataset of nouns and their genders created by Bergsma and Lin (2006). [26]

Jaidka et al. [94]) have discerned this absent facet within the realm of scientific document summarization and have undertaken its remediation by introducing a collaborative endeavor. This endeavor is designed to generate summaries that not only encapsulate the content within

the document's body but also encompass the broader perspective of the research community regarding these documents' evolution over time.

Regrettably, there has not been a single concerted effort to amalgamate these two approaches which would entail developing a summarization model that not only assimilates the content of the source document and its contextual background but also possesses the capability to gauge the article's influence on its respective academic community through an examination of the citations it has garnered. Considering this fact, in this work, we have introduced a standalone summarizer model which provides enriched summary from any scientific document combining the knowledge of the articles referenced in the body of the considering document plus the summary by means of summarizing the citation statements made on that particular article in other works. In pursuit of this goal, we have introduced a corpus for scientific document summarization that leverages the citation network. This corpus furnishes comprehensive information encompassing the document under scrutiny, the papers referenced within it, and the papers that have subsequently cited it. This corpus is a subset of the SSN corpus [10].

Another aspect of this work is that the introduced summarizer model has the ability to produce both the extractive and abstractive summaries in parallel. The rationale behind generating these two summaries in parallel lies in the reciprocal enhancement that occurs during the creation of each summary. The extractive summarizer represents a fusion of the heterogeneous graph-based neural network [227] and the graph attention network [232] and the abstractive summarizer is founded on a Longformer [22] decoder architecture. These two summarizer units establish a bidirectional information exchange by transmitting supplementary control signals to each other through the loss function. This coordinated approach ensures the concurrent generation of high-quality abstractive and extractive summaries. Prior to utilizing these two summarization units, the considered article is segmented using the segmentation technique proposed by Xing et al. [243] and for each segment it leverages the citation graph to incorporate background information. Subsequently, employing an hierarchical structure, the summaries of the segments are accumulated and the final summary is generated. Our contributions can be succinctly summarized as follows:

- We have developed a corpus, utilizing the citation network, for scientific document summarization containing 10k research articles. As per our knowledge, this is the first corpus curating the referenced and citing sides of the citation network for this task.

- We have developed a standalone model combing segmentation and summarization techniques that has the ability to gather background information form the reference articles and reflect the impact of the work on the corresponding research community considering the citations made on it while generating the summaries of the scientific document.

- The model has the capability to produce both the extractive and abstractive summaries in parallel. This parallel training of these two units allow each other to improve their individual performance.

7.2.2 Related Work

In the wake of the remarkable progress in neural network technology, a number of noteworthy research endeavors have emerged in recent years, focusing on the generation of extractive summaries [251] as well as abstractive summaries [257, 264] from the realm of scientific documents [50, 262].

Extractive summarizers identify pivotal sentences from the source document to form the summary; however, they tend to lack the coherent flow of information. Inceptive studies [61, 145] employ cosine similarity measurements between sentences for constructing a graph that encapsulates inter-sentence correlations. Certain contemporary research endeavors (e.g., [51, 252]) have incorporated discourse-related information from the articles in conjunction with inter-sentence correlations to formulate graphs and subsequently generate document summaries. Li et al. [120] fine-tuned T5 and integrated an extractive summarizer using Graph Convolutional Networks (GCN) for the purpose of generating summaries from extensive scientific documents. Wang et al. [227] have employed supplementary semantic units in graph neural network (GNN) to establish intricate relationships between sentences while designing their extractive summarizer. Cho et al. [48] have introduced a model (Lodoss) which segments the document and extracts the important sentences simultaneously.

Abstractive summarization lays significant emphasis on formulating a generalized summary, often requiring the utilization of sophisticated language generation models. These models are commonly built upon sequence-to-sequence (seq2seq) architectures, wherein the source document is treated as one sequence, while its corresponding summary is considered another. Cohan et al. [50] pioneered the development of the initial abstractive summarizer designed for lengthy scientific articles. Their approach incorporates a hierarchical encoder and a discourse-aware attentive decoder to accomplish this task. Mishra et al. [148] implemented a citation contextualization method to extract distinct and pertinent sentences from the document. Subsequently, they employed a multi-objective clustering approach to generate the final summaries. Liu et al. [126] harnessed the encoder-decoder framework of BERT, enabling their model BERTSUMABS to generate abstractive summaries. Wang et al. [231]entailed the independent extraction of latent topics from the input text, aiming to capture the underlying themes or concepts within the document. Subsequently, these extracted latent topics are employed to augment the performance of the summarizer. Yu et al. [257] utilized the guidance of an extractive

summarizer to enhance the performance of their abstractive summarizer (DimSum). It employs BART [115] as the foundation for its abstractive summarizer. The amalgamation of loss functions from both the extractive and abstractive summarizers contributes to the model's ability to generate improved lay summaries from scientific documents. Gupta et al. [81] employed both BERT and graph-based methodologies in their work on biomedical document summarization. PageSum [130] reduces memory overhead by treating the input document as a collection of pages based on locality. Each page is independently encoded by the abstractive model's encoder and the decoder generates local predictions for each page and assigns confidence scores to these predictions. HierGNN [174] is a neural encoder with reasoning capabilities, making it compatible for integration into various seq2seq neural summarization models.

A citation network has two sides: the articles being referenced in the considering literature, and the articles that have cited the considering article. To incorporate the information from the referenced articles while summarizing scientific documents, An et al. [10] introduced a substantial corpus, denoted as SSN, comprising 141,000 research papers interconnected through a citation network. Additionally, they presented a graph-based summarization model called CGSUM to extract information from both the source document and the citing texts, enhancing its summarization capabilities. Yasunaga et al. [251] introduced a corpus (CL-SciSumNet) comprising 1000 research articles with the citations made on them. The intention of their work is to generate summaries that also portrays the contribution of this work on the research community by means of accumulating the citing statements. However, as per our knowledge, there is no work still available that combines the information from the referenced articles to grasp the background knowledge, at the same time, portrays the impact of the work analyzing the citations made on that particular article while generating the summary. Filling up this gap has been the motivation of our work presented here.

7.2.3 Corpus Creation

Summarizing scientific literature is complex due to the need for contextual background knowledge, including references. Summarizer models require information from referenced articles. Additionally, assessing an article's true impact often requires analyzing citing statements. The SSN corpus offers background information from referenced articles, and the CL-SciSumNet corpus provides citing statements. However, there's no corpus connecting both facets of the citation network.

Considering these factors, we've introduced a corpus tailored for scientific document summarization. This corpus covers both sides of the citation network: the referenced articles and the citing statements. Our corpus is, in part, a subset of the SSN corpus. While the SSN corpus

contains background references, it lacks citing statements. To address this, we've enhanced our corpus by adding citing statements from citing papers to bridge this gap and build a more comprehensive corpus.

To create our corpus, we used the citation network to identify citing papers. We then manually extracted statements referencing the cited article from these citing papers. The SSN corpus, with its 141K articles, is quite extensive, so we selected a random subset of 10,000 papers for summarization. These papers have word lengths between 1,000 and 3,500 words, excluding background or related work sections. We deliberately chose this word length range to accommodate both the document and its citing statements within the Longformer's 4,096 token intake limit. In the papers earmarked for summarization, background and related work sections were removed. The dataset is partitioned into three subsets: 8,000 articles for training, 1,000 for validation, and another 1,000 for testing.

We have categorized the intentions expressed by citations into three classes: positive, neutral, or negative. For each paper, we have selected a maximum of 20 citation statements from each of these categories. Notably, negative citations are less prevalent, so for papers with limited negative citing sentences, we prioritized selecting more neutral and positive ones. To perform this categorization, we have experimented with various BERT-based models and ultimately fine-tuned RoBERTa RoBERTa [129] on the Citation Sentiment Corpus (CSC) [16] and used this model to classify all the curated citation statements.

To create summaries that amalgamate the perspectives of both the authors and the broader research community, we took a multi-step approach. Initially, we provided the abstracts of the research papers along with their corresponding citation statements to a fine-tuned T5 model [178]. This model had been trained on the CL-SciSumm corpus. It generated five different summaries for each document. Subsequently, these five summaries for each document were fed into a pre-trained Longformer architecture. This process produced five vector representations. To determine the most suitable summary, we compared these vector representations against the reference summary using cosine similarity. The summary with the highest cosine similarity to the reference summary was selected as our T5-Generated Summary, thereby reflecting a synthesis of both the authors' viewpoints and the broader research community's perspectives. To capture the background information we have used the citation network used directly in the SSN corpus. The maximum length of the summaries has been set to 500 tokens. For cleaning the equations and other unnecessary symbols, we have used the regex commands used by Singha Roy et al. [203].

To validate the quality of the proposed corpus, we have performed an analysis on a statistically representative sample of the corpus (95% confidence, and 3% error margin) with three human annotators' assistance. From the pool of 10,000 summarization samples, 400 were cho-

sen randomly and annotated by three annotators for this statistical analysis. Each annotator assessed whether the T5-generated summaries capture the same information as the abstract plus the citing statements. Annotator one said that 374 samples did, annotator two, 368 and annotator three, 371. Annotators one and two agreed that 368 samples compare correctly and 16 do not giving a Cohen's $\kappa = 0.89$. Between annotators two and three the agreement is with 396 samples (368 are correctly summarized and 28 are not) with $\kappa = 0.93$. Between annotators one and three the agreement is found for 398 summaries (370 correctly summarized and 27 not) with $\kappa = 0.94$.

7.2.4 Methodology

This section commences with the problem formulation, outlining how the summarization task of the considered document is enriched by utilizing the information contained in the network of referenced and citing papers. Then, the architecture of the proposed model is discussed.

7.2.4.1 Problem Formulation

Scientific papers possess a distinct attribute characterized by the presence of the citation relationship (referring to and referred to) among papers and the logical coherence in their content. Figure 7.2(a) visualizes this relationship augmented with the ideas of segmenting the considered paper and accumulating only the relevant sentences in the citing papers, both aspects which will be discussed later. These relationships will be used to enhance the effectiveness of summarization tasks in this domain.

To leverage this interconnected nature of scientific literature, we have utilized the concept of a citation graph. Description of this graph and its subgraphs will be used to describe how the model uses various portions of it. For the citation graph on the whole dataset $G = (V, E)$, each node $v \in V$ symbolizes a scientific article and each edge $e_{i,j} \in E$ portrays the relationship between articles represented by v_i and v_j. In this graph, the background knowledge for a scientific article v_i, (the papers to the left of the considered paper, $\mathcal{D}$, in Figure 7.2(a)), is represented by the subgraph G_i^{ref} which contains the relation between v_i and V_i^{ref} which is the set of the articles being referenced by v_i. This is further refined by another characteristic of the scientific article, the structured representation of its information [48]. To preserve this structure, we have applied the segmentation approach used by Xing et al. [243]. Our work applies this segmentation on document $\mathcal{D}$, the scientific article being considered, to define the citation subgraph ${G_i}^{Seg_p}$ for each segment Seg_p, $p = 1...n$ in $\mathcal{D}$ to accumulate the background information for all segments.

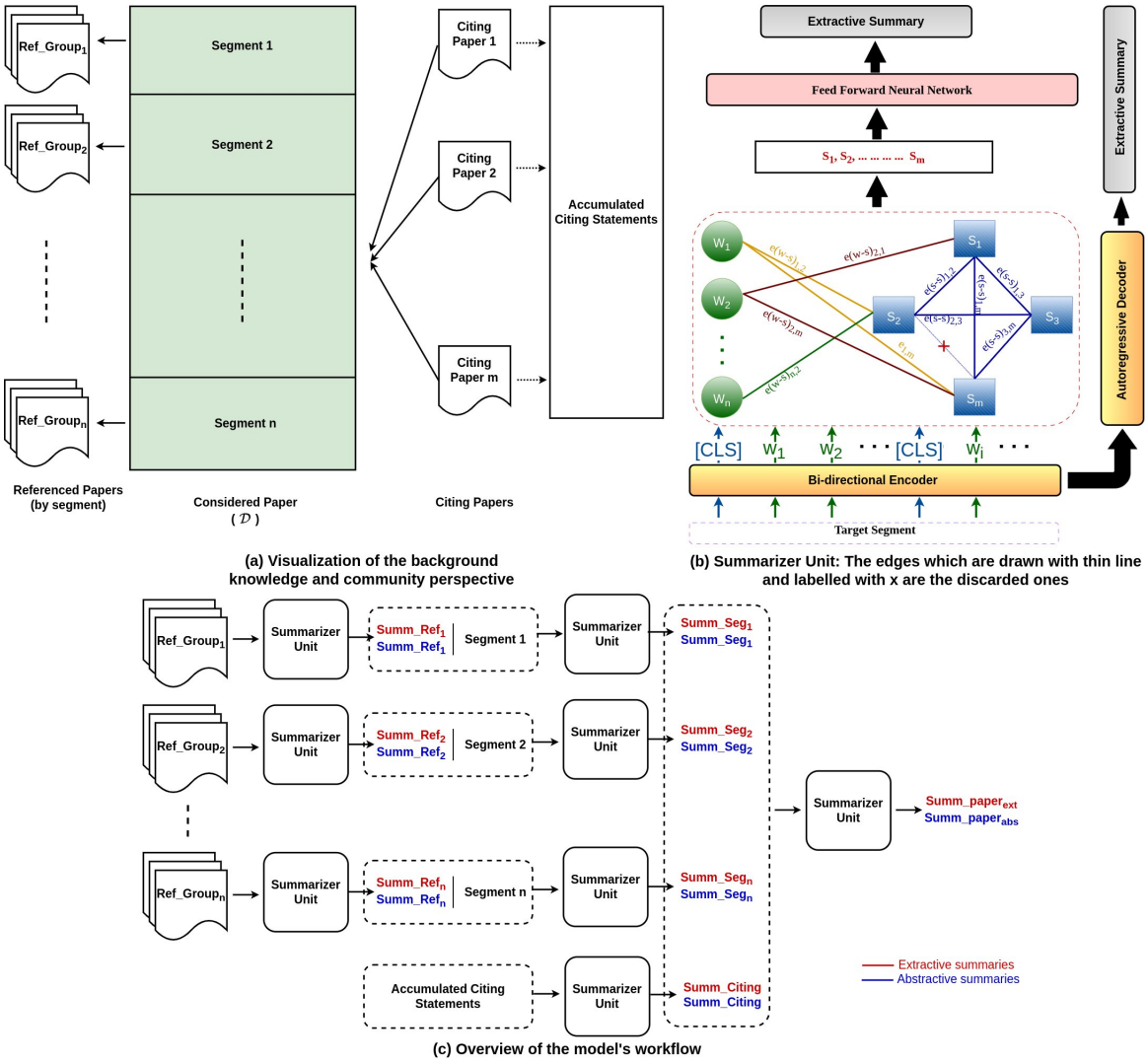

Figure 7.2: Architecture and workflow of the proposed model.

To define the second subgraph G_i^{citing}, we accumulate the citing statements referring to v_i (see Figure 7.2(a)) and use this as Seg^{citing}.

7.2.4.2 Model Architecture

Our proposed model operates concurrently on the segmentation and summarization tasks, enabling the acquisition of robust sentence representations. The model architecture is portrayed in Figure 7.2. Initially, the document is segmented into sections using the segmentation model introduced by Xing et al. [243]. This segmentation utilizes the word embeddings from Longformer as input and applies attentive Bi-LSTM on top of it to get the sentence representations. Another sentence representation is generated from pre-trained Longformer and these two features are concatenated together. This concatenated feature vector is then fed to the

following Bi-LSTM layer which predicts the section boundaries. This segmentation problem is formulated as a binary classification problem. To label each article, the sentence starting each segment of the article is labelled with 1 and all others with 0. This segmentation model is optimized with the binary cross entropy loss function (Equ. 7.15 where k is the number of sentences in the document).

$$\mathcal{L}_{seg} = -\sum_{i=1}^{k-1}[y_i \log p_i + (1 - y_i) \log(1 - p_i)] \tag{7.15}$$

After the segmentation is completed, the citation graph is utilized to aggregate the abstracts of the articles referenced by each segment p. For the considered document $\mathcal{D}$, represented by v_i in the citation graph, these articles are represented by the nodes in $G_i^{Seg_p}$. These groups of abstracts are then fed to the subsequent summarizer unit per segment.

The summarizer unit has two components: extractive and abstractive summarizer units. The architectural overview of the summarizer unit is depicted in Figure 7.2(b). When developing the extractive summarizer, we have focused on two discourse aspects: sentence-level semantic connections for information coherence and the influence of word-level semantics on sentence correlations. With these considerations, for a target document $\mathcal{D}$, we have designed the graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ where $\mathcal{V}$ symbolizes the nodes and $\mathcal{E}$ symbolizes the edges that connect these nodes. The set of nodes $\mathcal{V} = \mathcal{V}_w \cup \mathcal{V}_s$ where $\mathcal{V}_w = \{v_{w_1}, v_{w_2}, ..., v_{w_n}\}$ is the set of all the distinct words and $\mathcal{V}_s = \{v_{s_1}, v_{s_2}, ..., v_{s_m}\}$ denotes the set of sentences in $\mathcal{D}$, and $\mathcal{D}$ contains n unique words and m sentences. $\mathcal{E} = \mathcal{E}_{\mathcal{W}\text{-}\mathcal{S}} \cup \mathcal{E}_{\mathcal{S}\text{-}\mathcal{S}}$ is the edge weight matrix where $\mathcal{E}_{\mathcal{W}\text{-}\mathcal{S}}$ represents the edges between word and sentence nodes and each element $e_{w_i\text{-}s_j}$ in $\mathcal{E}_{\mathcal{W}\text{-}\mathcal{S}}$ is defined in such a way that $e_{w_i\text{-}s_j} \neq 0$ ($i \in \{1...n\}$, $j \in \{1...m\}$) if the sentence s_j contains the word w_i. $\mathcal{E}_{s\text{-}s}$ symbolizes the edges between sentences in the document. The sentence nodes are initialized with Longformer [CLS] tokens and the word nodes with:

$$w_i = \frac{\sum_{\forall s_j \in \mathcal{V}_s \wedge e_{w_i\text{-}s_j} \neq 0} vec_{w_{i,j}}}{\sum_{\forall s_j \in \mathcal{V}_s} |e_{w_i\text{-}s_j} \neq 0|} \tag{7.16}$$

where $|e_{w_i\text{-}s_j} \neq 0|$ is the number of occurrences of the word w_i in $\mathcal{D}$ and $vec_{w_{i,j}}$ symbolizes the Longformer word token for word w_i in sentence s_j. Each word-sentence edge $e_{w\text{-}s_{i,j}} \in \mathcal{E}_{\mathcal{W}\text{-}\mathcal{S}}$ is initialized with the corresponding TF-IDF value. Each cross-sentence edge $e_{s_x\text{-}s_y} \in \mathcal{E}_{\mathcal{S}\text{-}\mathcal{S}}$ is initialized with the cosine similarity between Longformer [CLS] tokens of sentences s_x and s_y.

Scientific articles contain a large number of sentences making operations on fully connected sentence node graphs computationally expensive. As a solution, we have discarded the edge connections between sentence nodes with cosine similarity values below a threshold, $\theta = 0.3$,

since experimentally, we have discovered that for $\theta \leq 0.3$ the summarization quality of the model is not affected. To further reduce the computational overhead, the vocabulary size is reduced by replacing words in the document with common synonyms.

Once the graph $\mathcal{G}$ has been constructed and initialized, a graph attention network (GAT) is applied over the word and sentence nodes in an iterative manner to update them. This GAT layer has been designed by following Wang et al. [227]. Considering, h_i as the hidden state representation of either $v_{w_i} \in \mathcal{V}_{\mathcal{W}}$ or $v_{s_i} \in \mathcal{V}_{\mathcal{S}}$ where $h_i \in \mathbb{R}^{d_h}$ and $i \in \{1, ..., (n+m)\}$, the GAT layer (incorporating the edge information) is delineated as:

$$\mu_{i,j} = LeakyReLU(\omega_a[\omega_q h_i; \omega_k h_j; e_{i,j}]) \tag{7.17}$$

$$\alpha_{i,j} = \frac{\exp(\mu_{i,j})}{\sum_{l \in \mathcal{N}_i} \exp(\mu_{i,l})} \tag{7.18}$$

$$u_i = \sigma(\sum_{j \in \mathcal{N}_i} \alpha_{i,j} \omega_v h_j) \tag{7.19}$$

where, ω_v ω_k, ω_q, and ω_a are weight matrices that are updated iteratively. $\mathcal{N}_i$ is the set of 1-hop distant neighbour nodes. The attention value between neighbour nodes h_i and h_j is depicted by $\alpha_{i,j}$. For $\mathcal{K}$ attention heads, this GAT layer is designed as:

$$h'_i = \|_{k=1}^{\mathcal{K}} \sigma(\sum_{j \in \mathcal{N}_i} \alpha_{i,j}^k \omega^k h_i) \tag{7.20}$$

Furthermore, a residual connection has been added to prevent gradient vanishing and the final hidden representation, h_i, is:

$$h_i = h'_i + h_i \tag{7.21}$$

In the first step of model training, the sentence nodes are updated, influenced by their 1-hop distant word nodes, using the aforementioned GAT layer and the position-wise feed-forward network (FFN) [227]:

$$\mathcal{U}_{w \to s}^{(1)} = GAT(\mathcal{H}_s^{(0)}, \mathcal{H}_w^{(0)}, \mathcal{H}_w^{(0)}) \tag{7.22}$$

$$\mathcal{H}_s^{(1)} = FFN(\mathcal{U}_{w \to s}^{(1)} + \mathcal{H}_s^{(0)}) \tag{7.23}$$

where $\mathcal{H}_w^0 = \mathcal{V}_w$, and $\mathcal{H}_s^0 = \mathcal{V}_s$. In Equ. 7.22, $\mathcal{H}_s^0$ has been considered as the attention query matrix, and $\mathcal{H}_w^0$ as both the key and value matrices.

Once the sentence nodes are updated using the adjacent word nodes, in the following step

the sentence nodes are updated using cross-sentence correlations, followed by a word node update step using the last-modified sentence node representations. Thus, each iteration comprises a sequence of sentence-sentence, sentence-word, word-sentence and cross-sentence edge updates. At the t^{th} iteration, this operation can be stated as:

$$\mathcal{U}_{s\to s}^{(t+1)} = GAT(\mathcal{H}_s^{(t)}, \mathcal{H}_s^{(t)}, \mathcal{H}_s^{(t)}) \tag{7.24}$$

$$\mathcal{H}_s'^{(t+1)} = FFN(\mathcal{U}_{s\to s}^{(t+1)} + \mathcal{H}_s^{(t)}) \tag{7.25}$$

$$\mathcal{U}_{s\to w}^{(t+1)} = GAT(\mathcal{H}_w^{(t)}, \mathcal{H}_s'^{(t+1)}, \mathcal{H}_s'^{(t+1)}) \tag{7.26}$$

$$\mathcal{H}_w^{(t+1)} = FFN(\mathcal{U}_{s\to w}^{(t+1)} + \mathcal{H}_w^{(t)}) \tag{7.27}$$

$$\mathcal{U}_{w\to s}^{(t+1)} = GAT(\mathcal{H}_s'^{(t+1)}, \mathcal{H}_w^{(t+1)}, \mathcal{H}_w^{(t+1)}) \tag{7.28}$$

$$\mathcal{H}_s^{(t+1)} = FFN(\mathcal{U}_{w\to s}^{(t+1)} + \mathcal{H}_s'^{(t+1)}) \tag{7.29}$$

$$\forall e_{s_i - s_j} \in \mathcal{E}_{\mathcal{S}-\mathcal{S}} = \cos(\mathcal{H}_{s_i}^{(t+1)}, \mathcal{H}_{s_j}^{(t+1)}) \tag{7.30}$$

The Longformer decoder has been utilized as the abstractive summarizer following the approach used by Yu et al. [257].

For each segment, once the abstracts of the referenced articles are extractive- and abstractive-summarized, these two summaries are individually concatenated with the segment. These texts are then fed to their corresponding summarizer unit to produce extractive and abstractive summaries of each segment. At this step of the hierarchy, the accumulated citing statememts are also extractive- and abstractive-summarized. In the last hierarchical step, the extractive and abstractive segment summaries are concatenated with the corresponding summary of the citing statements and fed to the corresponding summarizer unit to produce the final extractive and abstractive summaries of the considered article. Both the extractive and abstractive summarizer units use cross-entropy loss functions ($\mathcal{L}_{ext}$ and $\mathcal{L}_{abs}$, accordingly). The model's loss function, $\mathcal{L}$ is defined as:

$$\mathcal{L} = \mathcal{L}_{ext} + \mathcal{L}_{abs} + \mathcal{L}_{seg} \tag{7.31}$$

7.2.5 Experiments

This section first gives a brief description of the hyper-parameter settings and hardware used for the model implementation and then presents the results achieved by the proposed model described in the previous section on the corpus outlined in Section 7.2.3.

The experiments have been conducted on a 48GB NVIDIA RTX A6000 GPU with *batch size* = 5 to accommodate the large number of sentences in the scientific documents. For model

Table 7.4: Results on the proposed corpus. The results consider both the abstracts and the T5-generated summaries incorporating citation statements as the reference summaries.

Models	On Abstracts as Summaries				On T5-Generated Summaries			
	R-1	R-2	R-L	METEOR	R-1	R-2	R-L	METEOR
Extractive								
BERTSumExt [126]	45.63	15.99	41.91	34.89	46.01	16.17	42.18	34.97
HeterSumGraph [227]	46.35	16.22	42.64	35.02	46.81	16.29	42.82	35.16
CGSUM [10]	46.98	17.02	44.17	38.26	46.96	16.96	43.85	37.93
Lodoss [48]	47.17	17.22	44.37	38.61	47.29	17.24	44.47	38.66
Proposed Model (Extractive)	48.39	18.18	45.18	39.13	48.43	18.21	45.19	39.11
Abstractive								
PTGen+Cov [197]	43.99	14.12	38.16	33.51	43.97	14.10	38.18	33.46
BERTSumAbs [126]	45.01	15.33	38.96	34.59	45.02	15.36	39.00	34.64
BERT+CopyTransformer [8]	45.62	15.78	39.93	34.84	45.54	15.81	39.91	34.88
Proposed Model (Abstractive)	48.12	17.96	44.91	38.85	48.04	17.99	44.71	38.82

training with a small batch size, we have followed the approach of Sefid et al. [198]. Gradients are collected for ten steps and then the parameters are adjusted. The NOAM scheduler is used to regulate the learning rate. Furthermore, to prevent the exploding gradient problem, we have used gradient clipping. The extractive summarizer is initialised with 768-dimensional Longformer embeddings. After that, the extractive summarizer unit uses the GAT (with 8 attention heads) and the following FFN layer to update the graph nodes. After every forward pass, the abstractive and extractive summarizer units' losses are calculated separately. If either unit's validation loss decreases for 5 continuous epochs, its parameter values are stored and its training is paused for the next 10 iterations. We have trained the model for 200 iterations. The FFN hidden layer size is set to 512. For the parallel training of the summarizers, we have followed the approach proposed by Yu et al. [257]. For the segmentation model, apart from the word embedding dimension, we have replicated the hyper-parameter settings used by Xing et al. [243]. This model is fed with 768-dimensional Longformer word vectors.

We have assessed the segmentation performance using F-1 scores. Like Cho et al. [48], we have experimented with predicting the first sentence and last sentence of each segment and found that when predicting the first sentence of each segment, the model performs better which supports the claim in [48]. With the joint training of segmentation and summarization, our segmentation model has achieved 86.19 F-1 score on the segmentation task when predicting the first sentences of the segments. We have also noticed that sentences near the segment boundaries are more prone to be included in the summaries.

In order to assess the efficacy of our model for extractive summarization, we undertake the training and evaluation of the following extractive summarization models with our adapted corpus: (1) BERTSumExt [126], an exemplar grounded in BERT; (2) HeterSumGraph [227], a heterogeneously structured graph-based technique that accounts for inter-sentence relation-

ships by incorporating supplementary semantic elements; (3) CGSUM [10], a graph-based summarization model that incorporates the information from the source paper plus the referenced articles; and (4) Lodoss [48], it performs the segmentation and summarization tasks in parallel regularized by the determinantal point processes regularizer. In the context of abstractive summarization benchmarking, our experimentation encompassed the utilization of the following models: (1) PTGen+Cov [197], founded upon a hybrid pointer generator network designed to facilitate verbatim transcriptions from the source text; (2) BERTSumAbs [126], a model rooted in the BERT architecture; and (3) BERT+CopyTransformer [8], which leverages BERT-windowing techniques to manage textual content exceeding the inherent BERT window limitations. While training these models, to incorporate the background information, we have concatenated the abstracts of the referenced articles and the considered article following An et al. [10]. The citation statements are also concatenated at the end. The same approach is used for HeterSumGraph and CGSUM. To overcome the token intake limitation of the BERTSumEXT and BERTSumAbs, we have added additional positional encoding which is added randomly and fine-tuned in the training phase [10].

The performances for the prior models and our novel proposals are presented in Table 7.4 using four commonly used metrics. For reference summaries, we have taken into account not only the paper abstracts but also the summaries that we have produced by amalgamating the abstracts with the citing statements via the T5 framework.

HeterSumGraph scrutinizes immediate associations among words and sentences within textual contexts limited to a maximum of 50 sentences. Conversely, our innovative model not only takes into account these immediate cross-sentence correlations but is also adept at handling more extensive text spans, accommodating up to 3500 words. Over the sentence-word relationships presented in HeterSumGraph, our model provides inter-sentence correlations. These supplementary functionalities, coupled with the enhanced word and sentence features offered by LongFormer, collectively contribute to a notable enhancement in our model's performance. CGSUM can take up to two-hop reference articles. For the experiment here, it has been restricted to one-hop to comply with our proposed corpus. However, CGSUM considers all the abstracts from the reference article at once, rather than being used segment by segment. Using reference abstracts segment by segment and utilizing an hierarchical summarization approach over segments allows our model to benefit from the background information in the reference articles where it is needed. However, it is essential to acknowledge that the heightened capabilities of our model necessitate a commensurate increase in computational time and resource allocation. In terms of performance, our model demonstrates a substantial gain over other models for the extractive summarization task. The extractive summarizer unit, in our model, has achieved 45.18 Rouge-L (R-L) and 39.13 METEOR scores over the "abstracts as summaries"

which is 0.81 R-L and 0.52 METEOR higher than Lodoss, which is the best performing model among the considered other extractive summarizers . Over the "T5 generated summaries", our model has outperformed Lodoss by 0.72 R-L and 0.45 METEOR scores by attaining 45.19 R-L and 39.11 METEOR scores. Like the extractive summarizer unit, our abstractive summarizer unit has also outperformed the other considered abstractive summarizer units by attaining 44.91 R-L and 38.85 METEOR scores over the "abstracts as summaries", and 44.71 R-L and 38.82 METEOR scores over the "T5 generated summaries". The best performing model among the considered abstractive summarizers, BERT+CopyTransformer, has achieved 39.93 R-L and 34.84 METEOR over the "abstracts as summaries", and 39.91 R-L and 34.88 METEOR over the "T5 generated summaries".

To perform the ablation study, different units from the proposed model are discarded and then the performances are reported (see Table 7.5). Experimental results show, if the word-sentence update step is discarded, the model is affected more than by discarding the sentence-sentence update step. This difference corresponds with our knowing that the sentence nodes are still connected via the word nodes, and suggests that removing the word-sentence update step has a greater information loss. Furthermore, the results show that replacing uncommon words with corresponding common synonyms not only reduces the computational burden, but also improves the performance and justifies the claim by Wang et al. [227] which states that articles containing words with higher node degree not only make the summarization task easier for the deep learning models but also improves the performance. Another observation that we have drawn from the ablation study is that discarding the extractive summarizer affects the abstractive summarizer more than the extractive summarizer gets affected when the abstractive summarizer unit is discarded. These performance drops for the summarizer units also indicate the significance of the parallel training of the extractive and abstractive summarizers. Both the extractive and abstractive summarizer units are affected with a performance drop in both cases when the background information provided by the citation graph or the segmentation units are discarded. It proves that providing background information segment-by-segment rather than providing this information altogether helps the summarizer model attain better performance.

7.2.6 Conclusion

In this paper, we have introduced a scientific document summarization model that leverages references within the article to provide background information and reflects the impact of the cited work on the research community through citation statements. We have created a novel corpus based on a citation graph, encompassing abstracts of reference papers and citing statements for 10,000 scientific articles. This work takes the background information from the

Table 7.5: Ablation Study on the T5 generated summaries: † indicates the extractive summaries and ∗ indicates the abstractive summaries.

Discarded Unit	R-L	METEOR
Sentence-Sentence update†	43.98	38.68
Word-Sentence update†	42.51	37.22
Abstractive summarizer†	43.95	38.47
Extractive summarizer∗	41.63	36.56
Citation network†	42.17	37.16
Citation network∗	41.74	36.89
Segmentation unit†	43.21	38.14
Segmentation unit∗	42.68	37.79
Synonym replacement†	44.07	38.25
Synonym replacement∗	42.94	37.58

reference articles segment-by-segment. As per our knowledge, this is the first approach to bridge the gap between two facets of the citation graph in scientific document summarization.

7.2.7 Limitations

We have trained both the extractive and abstractive summarizer units for a large number of epochs. Though to prevent any unit from being over-fitted we have checked the curve of validation loss after every 5 epochs. This is very computationally expensive and demands a longer period of time for model training.

7.3 Investigating Semantic Similarity-Induced Parallel Training of Abstractive and Extractive Scientific Document Summarizers

This section is based on the paper titled "Investigating Semantic Similarity-Induced Parallel Training of Abstractive and Extractive Scientific Document Summarizers" co-authored with Robert E. Mercer. Currently, this paper is under preparation for conference paper submission.

Scientific document summarization focuses on condensing scientific literature, research papers, or technical documents into concise summaries while preserving crucial scientific concepts, findings, and conclusions. In this work, we have presented an approach to improve the performance of the summarization models using a parallel training of the extractive and abstractive summarizers together with a modified loss function. The modified loss function is a union of cross-entropy loss and semantic similarity among the generated and reference sum-

maries. The experiments that are used to validate the parallel training method and new loss function have used a combination of four recently state-of-the-art extractive summarizers and four abstractive summarizers. Results indicate that for all combinations, both the extractive and abstractive summarizers gain significant performance boosts. It is conjectured that the parallel training method and new loss function will improve any combination of quality extractive and abstractive summarizers.

7.3.1 Introduction

Document summarization refers to the process of condensing a written record or collection of records into a concise and coherent synopsis while maintaining the crucial details and primary concepts of the original document. It entails examining the content of the original document(s) and extracting the most pertinent and significant information to produce a shorter version that captures the fundamental meaning and noteworthy points of the source material. The objective of document summarization is to offer a comprehensive overview of the original text(s), allowing readers to quickly comprehend the principal ideas and extract pertinent information without the need to peruse the entire document(s).

With recent advancements in neural networks and large pre-trained language models [56, 258, 22], researchers have made significant progress in the field of short document summarization like news article summarization, typically dealing with documents of approximately 650 words [154, 45, 156, 126] However, these models face challenges when processing longer texts such as scientific papers. Scientific papers can range from 2,000 to 7,000 words in length, and the corresponding summary, which is usually the abstract, is expected to be more than 200 words, as opposed to the concise 40-word summaries found in news headlines [261].

Additionally, for long documents, it is imperative to uphold the organizational structure provided by chapters, sections, headings, and bullet points. This allows readers to effortlessly navigate the document and locate the most important information and key details encapsulated within its contents [47]. This section-oriented representation of texts creates another challenge for the summarizer models as information is organized in sections, not sequentially. Furthermore, the computational complexity of attention in Transformer-based models, as introduced by Vaswani et al. [223], is quadratic with respect to the length of input tokens. This quadratic complexity poses a significant challenge and renders these models impractical for certain applications [261].

Scientific document summarization research focusses on two types of summaries: extractive and abstractive. Extractive summarization refers to the process of creating a concise summary of a document by directly extracting the most important and relevant sentences from

the original text. Rather than generating new sentences like abstractive summarizers, extractive summarization identifies sentences from the source document that best capture the key ideas and main points. This approach allows for a more direct representation of the original document's salient information in the summary. Initially, RNN-based models [273, 154, 45] have been explored for this task. However, these models fail due mostly to the inability of the RNN models to capture dependencies between long-distant sentences. Recently, fine-tuned pre-trained language models (PLMs) have been commonly used for the extractive summarization task. However, PLMs focus on sequential context by incorporating linear positional encoding to input token embeddings, but they do not explicitly consider hierarchical text structure information [189]. Addressing these issues, in the most recent approaches, extractive summarizers are considering section information [189], graph-based approaches to incorporate topic information [242], location of the keywords in the text [261], and heterogeneous relationship between words and sentences [227].

Unlike extractive summarization, which involves selecting and rearranging existing sentences, abstractive summarization entails comprehending the meaning of the input text and generating sentences that effectively capture the essential information in a more natural fashion. Different variations of sequential neural network architectures have emerged as the predominant approach in abstractive summarization [115, 197]. Despite this progress, machine-generated summaries still fall significantly short in terms of quality when compared to human-generated summaries [174, 90]. Just like the extractive summarizers, abstractive summarizers with sequential neural network lack the ability to capture long-distant sentence dependencies and hierarchical structure present in the long text documents. Additionally, in these models, the self-attention module brings about a quadratic increase in memory requirements as the length of the input sequence grows [130]. This is why recent research has started incorporating knowledge of hierarchical structure of the document [174], attention over the locality of the text [130], and adding auxiliary extractive salience [228].

One contribution described here is a training method that trains an extractive and an abstractive summarizer in parallel. To provide strong evidence that this training method is summarizer agnostic, we have experimented with four recently state-of-the-art (SOTA) extractive and abstractive summarizers. In all possible combinations, we have trained these models with the proposed training approach, and the results indicate that in every case, both the extractive and abstractive summarizers have performance boosts.

Having the extractive, abstractive, and reference summaries during training provides another avenue for performance improvement. While current research focusses on the Rouge metric to measure model performance, another important aspect of summary quality is the preservation of the semantics of the summarized document. One common issue for all of the

SOTA models being used in the experiments is that while generating the summaries, they miss taking advantage of summary-level semantic similarity (GRETEL [242] uses semantic similarity at the sentence level). One key approach is to make the generated summaries more semantically similar to the reference summaries. Since the parallel training method generates two summaries, a further approach is to push the summaries to be more semantically related. To address this issue, we have modified the typical cross-entropy loss function to include semantic similarity of the summaries using cosine difference.

Joining these two novel ideas, we provide a semantic similarity-induced parallel training of extractive and abstractive summarizers where the loss function guides the individual summarizer units to move closer to the reference summary on the basis of their underlying semantics.This new semantically enhanced parallel method shows significant performance gains for all combinations of the SOTA extractive and abstractive summarizers, strongly suggesting that it is model agnostic, thereby allowing us to conjecture that this will be the case for any combination of quality summarizers.

7.3.2 Related Work

Due to the encouraging advancements in short document summarization, recently there has been a growing research interest in long document (such as scientific articles) summarization in both extractive [227, 189, 242] and abstractive [8, 228, 174] manners.

The goal of **Extractive Text Summarization (ETS)** is to categorize sentences in a document using labels that indicate whether a particular sentence should be included in the summary. Most recent ETS models [261, 189] for long documents are based on transformer-based architectures [22, 258] as they have the ability to work with longer sequences in comparison to the RNN-based models. Liu et al. [126] have introduced BERTSUMEXT, a method that fine-tunes BERT by incorporating stacked Transformer layers and a sigmoid classifier. Instead of using the standard Transformer encoder for document encoding, Zhang et al. [264] have proposed HIBERT, a hierarchical Transformer encoder that includes a sentence encoder and a document encoder. They have pre-trained this encoder and then fine-tune it specifically for the ETS task. Zhong et al. [271] utilizes the siamese-BERT architecture to select candidate extractive summaries by means of computing the semantic similarity between the candidate and reference summaries. Recently, state-of-the-art extractive summarizers for the scientific documents are HiStruct+ [189], GRETEL [242], HEGEL [261], and Lodoss [47].

HiStruct+ [189] involves formulating, extracting, encoding, and explicitly injecting hierarchical structure information into an extractive summarization model to incorporate both local and global contextual information and is based on a pre-trained Transformer language model

that focuses solely on encoding following the concept of BERTSUMEXT [126]. . The major contribution of this work is the introduction of hierarchical positional encoding of sentences which helps the model to integrate hierarchical information in the PLMs for the summarization task.

GRETEL [242] combines the graph contrastive topic model with a PLM to maximize the utilization of both global and local contextual semantics for ETS of long documents. It integrates a hierarchical transformer encoder and graph contrastive learning to effectively capture and incorporate global semantic information from the overall document context and the desired summary. GRETEL aims to encourage the model to extract pertinent sentences that are topically relevant to the gold standard summary, while minimizing the inclusion of redundant sentences that cover sub-optimal topics. One of the key aspects of GRETEL is its emphasis on converging to topic representation of documents and sentences that exhibit high semantic similarity with the gold summary.

HEGEL [261] have introduces a hyper-graph transformer layer to capture high order cross-sentence relationships from a long document. Various types of sentence dependencies, such as latent topics, keyword coreference, and section structure, are incorporated in order to enhance the summarization process. It represents a document as a hyper-graph where an edge can connect to any number of vertices. Each dependency is represented by a one hot matrix and finally they are concatenated to form an incidence matrix which provides additional information to make a connection between sentences containing the same topic, keywords, or other dependencies, even if these sentences occur in different sections of the document.

Lodoss [47] adopts a novel strategy by simultaneously learning robust sentence representations through both summarization and document segmentation. This integrated process allows Lodoss to capture the essence of the document effectively by recognizing document structure along with encoding salient content. This is further enhanced by incorporating an optimization regularizer (based on determinantal point process) that encourages the diversity in selecting candidate sentences and avoiding redundancy. This model is architectured on top of Longformer [22] following a stacked double-layered inter-sentence transformer.

Another type of summarization, **Abstractive Text Summarization (ATS)**, aims to generate summaries that contain new sentences that are not directly extracted from the source text. Unlike extractive summarization, which selects and rearranges existing sentences, ATS aims to generate concise and coherent summaries by generating novel sentences that capture the essence of the source text. This process involves understanding the source text, generating new content, and ensuring the generated summary is coherent and informative. Liu et al. [126] utilizes the encoder-decoder framework of BERT which enables BERTSUMABS to generate abstractive summaries by leveraging the encoded information and generating new sentences

that capture the gist of the document. Wang et al. [231] propose a two-step approach for improving the summarization model. The first step involves extracting latent topics independently from the input text to capture the underlying themes or concepts within the document. In the second step, these extracted latent topics are utilized to enhance the performance of the summarization model. Aralikatte et al. [11] have used neural topic modeling with bag-of-words as input features and then implemented transformer-based encoder-decoder architecture for generating abstractive summaries. Fu et al. [68] have examined the extraction of topic distributions at both the document and paragraph levels. These distributions then have been used as guidance in the abstractive summarization process. Yu et al. [257] have employed the guidance of an extractive summarizer to improve the performance of the abstractive summarizer (DimSum). For the abstractive summarizer, DimSum utilizes BART [115]. The combined loss function of the extractive and abstractive summarizers helps the model to generate better lay summaries from scientific documents. Recently, the state-of-the-art models for ATS are DYLE [138], FACTORSUM [66], PageSum [130], and HierGNN [174].

DYLE [138] introduces a dynamic latent extraction mechanism that involves training both an extractor and a generator simultaneously. To determine the probability of an output token, DYLE calculates it based on each input snippet independently, while the generation probability is determined by the generator's dynamically assigned weights and previously generated tokens. The extractor is optimized using two surrogate losses: the extractive oracle, which uses a greedy search to find the best ROUGE scores to serve as targets for the extractor, and the consistency loss, which encourages the extractor to move its predicted weights toward the averaged dynamic weights predicted by the generator.

FactorSum [66] is built on the premise that separating content selection from the allocation of resources to cover important content enhances the effectiveness and versatility of abstractive summarization systems. This model achieves the disentanglement of content selection and resource allocation by employing an energy function that factorizes the summarization process into two distinct steps. During the initial step, FactorSum generates abstractive summary views that specifically cover the most significant information found within subsets of the input document. In the second step, FactorSum combines the generated summary views into a final summary while adhering to a budget and content guidance. This guidance can originate from various sources, such as an advisor model like BART [115] or BigBird [258].

PageSum [130] is built on the concept of leveraging locality to reduce memory overhead, while still providing informative summaries. Rather than treating the input document as a single uninterrupted sequence, PageSum represents it as a collection of pages, constructed based on the concept of locality. Each page is encoded independently by the encoder of the abstractive model. During decoding, the decoder generates local predictions for each page and

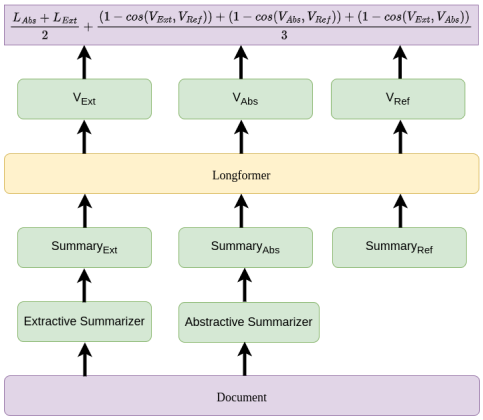

Figure 7.3: Parallel training process of the extractive and abstractive summarizers with the semantic similarity loss function.

assigns confidence scores to these predictions. These local predictions and confidence scores are then combined to produce the final outputs. Notably, tokens from different pages do not interact directly with each other during the encoding and decoding processes, emphasizing the significance of locality in text summarization.

HierGNN [174] is a neural encoder that integrates a reasoning capability, making it suitable for integration into any sequence-to-sequence (seq2seq) neural summarization model. The HierGNN model initially learns a latent hierarchical graph using a sparse variant of the matrix-tree computation technique [128, 110]. Next, it formulates sentence-level reasoning as a graph propagation problem by employing a novel message passing mechanism. During the decoding process, a graph-selection attention mechanism acts as a source sentence selector, hierarchically guiding the attention module to focus on specific tokens in the input sentences and generate more precise summaries.

7.3.3 Semantic Similarity-induced Parallel Training of Extractive and Abstractive Summarizers

The training methodology employed in this study adopts a parallel approach, employing a siamese structure that incorporates both an extractive and an abstractive summarizer. To enhance their performance, a modified loss function is utilized. Figure 7.3 illustrates the overall architecture. The primary focus of this work is not to propose a new model, but rather to introduce a training approach for the summarizers, incorporating a semantic similarity loss function to improve their performance. This training approach serves as a bidirectional guidance mechanism, aiming to prompt both summarizers to generate summaries that are more semantically similar to the reference gold-standard summaries.

The parallel training of the summarizers is a straightforward approach. At first, the extractive and abstractive summarizers try to generate the summaries of a given document. Once both the extractive ($Summary_{Ext}$) and abstractive ($Summary_{Abs}$) summaries are generated, they are fed to the following Longformer [22] layer along with the reference gold-standard summary ($Summary_{Ref}$). This Longformer layer is responsible for generating the vector representations of the summaries (V_{Ext}, V_{Abs}, V_{Ref} for the extractive, abstractive, and reference summaries, accordingly). In the following step, the overall loss of the model is computed using our semantic similarity loss function.

The semantic similarity loss function has two parts. The first part utilizes the cross-entropy loss which is the commonly used approach to improve the Rouge values of the generated summaries. Following the work of DimSum [257], this portion considers the cross-entropy of both the extractive (L_{Ext}) and abstractive (L_{Abs}) summarizers by taking the summation of them. The second portion of the loss function, considers the semantic aspect of the generated summaries. This portion of the loss function is three-fold. The first fold computes the distance between the extractive and reference summaries at the semantic space. To measure the semantic distance between the extractive and reference summaries, the cosine function $((1 - \cos(V_{Ext}, V_{Ref})))$ is used. It tries to guide the the extractive summarizer to generate summaries semantically similar to the reference summary. The second fold $((1 - \cos(V_{Abs}, V_{Ref})))$ does the very same job for the abstractive summarizer. The last fold $((1 - \cos(V_{Ext}, V_{Abs})))$ tries to make both the summarizer units semantically similar. It is used basically to push the low-performing summarizer model towards the gold standard. Finally, these three last-mentioned folds are normalized so that the overall loss doesn't become too big. The final semantic similarity loss function (L) is described as:

$$L = ((1 - \cos(V_{Ext}, V_{Ref})) + (1 - \cos(V_{Abs}, V_{Ref})) + (1 - \cos(V_{Ext}, V_{Abs})))/3 + (L_{Abs} + L_{Ext})/2 \tag{7.32}$$

Although this work draws motivation from Dimsum [257] and MatchSum [271], the major difference with these works is the introduction and incorporation of the semantic similarity loss function for the parallel training of the extractive and abstractive summarizers. On top of that, unlike Dimsum, which primarily aims to guide the abstractive summarizer by training it alongside the extractive summarizer, our approach provides bidirectional guidance to each summarization component. In contrast to MatchSum, which focuses on constructing an extractive summarizer based on the semantic similarity between candidate summaries and the reference summary, our work concentrates on enhancing the performance of existing summarizers (both extractive and abstractive) through this novel training method.

7.3.4 Experimental Setup and Analysis of Results

7.3.4.1 Corpus Description

In order to evaluate the performance of our semantic similarity-induced training approach, we have conduct experiments on two benchmark corpora: PubMed and arXiv [50]. These datasets consist of research articles paired with their respective abstracts. In our study, we treat the abstracts as the reference summaries. Table 7.6 outlines their key characteristics. The default train/validation/test set split proposed by Cohan et al. [50] is employed for these two corpora in our study.

Table 7.6: Statistics of the PubMed and arXiv datasets.

Datasets	Number of documents	Avg. word count per document	Avg. summary length
PubMed	133	3016	203
arXiv	215K	4938	220

7.3.4.2 Experimental Results Analysis

Here, we present the results gathered from the series of experiments conducted using our semantic similarity-induced parallel training approach. We demonstrate the progressive enhancement in performance exhibited by the summarizer models to illustrate the significance of each component.

For the experiments we have considered four SOTA extractive and four SOTA abstractive summarizers. These models have been trained in 16 combinations, pairing one extractive summarizer with one abstractive summarizer for each combination. The four extractive summarizers are: HiStruct+ [189], HEGEL [261], GRETEL [242], and Lodoss [47]. The four abstractive summarizers are: DYLE [138], FACTORSUM [66], PageSum [130], and HierGNN [174].

Tables 7.7 and 7.8 present the performance of the extractive and abstractive summarizer models, respectively, when trained individually on the two benchmark datasets.

Table 7.9 displays the results obtained when the summarizers are trained in parallel with the loss function used in DimSum [257]. This loss function combines the cross-entropy losses of the extractive and abstractive summarizers. Comparing the performances of the summarizer models in Tables 7.7, 7.8, and 7.9, it is clear that the parallel training approach improves the summarizer units' performances.

Table 7.10 contains the performance of the summarizer units when as semantic units only the cosine distances between the reference and extractive summaries and the reference and

Table 7.7: Rouge scores for four recently state-of-the-art extractive summarizers on the PubMed and arXiv corpora.

Extractive	PubMed			arXiv		
Model	R-1	R-2	R-L	R-1	R-2	R-L
HiStruct+ [189]	46.59	20.39	42.11	45.22	17.57	40.16
HEGEL [261]	47.13	21.00	42.18	46.41	18.17	39.89
GRETEL [242]	48.20	21.20	43.16	48.17	20.31	42.84
Lodoss [47]	49.38	23.89	44.84	48.45	20.72	42.55

Table 7.8: Rouge scores for four recently state-of-the-art abstractive summarizers on the PubMed and arXiv corpora.

Abstractive	PubMed			arXiv		
Model	R-1	R-2	R-L	R-1	R-2	R-L
DYLE [138]	46.22	17.13	41.55	46.41	17.95	41.54
FACTORSUM [66]	47.50	20.33	43.76	49.32	20.27	44.76
PageSum [130]	48.73	21.33	44.67	49.72	20.98	44.69
HierGNN [174]	49.62	21.74	45.32	49.88	20.81	44.84

abstractive summaries are considered as part of the loss function.

Table 7.11 provides the Rouge scores achieved by the extractive and abstractive summarizers when the loss function considers the semantic distance between the generated extractive and abstractive summaries along with the reference-extractive and reference-abstractive semantic distances.

Results in Tables 7.9, 7.10, 7.11 show the gradual performance boosts of the models. As an example, the performance boosts gained by HiStruct+ (the poorest performing Extractive summarizer) for the PubMed corpus in terms of Rouge-L are 0.07, 0.09, 0.1, and 0.12 (averaged for the Extractive and Abstractive results in Table 7.9) over the original HiStruct+ performance on the PubMed corpus when it is coupled with DYLE, FACTORSUM, PageSum, and HierGNN, respectively, for the parallel training method only. When the semantic similarity loss function is coupled with it, the performance boosts rise to 0.38, 0.42, 0.45, and 0.51, maintaining the same order (see Table 7.11). From Table 7.10, we can see that the performance boost is slightly lower when the semantic distance between the generated extractive and abstractive summaries is not included in the loss function. For Lodoss, the best performing extractive summarizer, the performance boosts (average of the Extractive and Abstractive scores for Rouge-L for the PubMed corpus) are 0.60, 0.67, 0.71, 0.79, respectively, when the model is trained with the full semantic similarity loss function and parallel training. For only the parallel training, these improvements are 0.14, 0.19, 0.23, 0.24, respectively. A similar pattern of results is observed for the other investigated extractive summarizers for both corpora, as well.

Table 7.9: Rouge scores for the sixteen combinations of four recently state-of-the-art extractive and abstractive summarizers on the PubMed and arXiv corpora using only the parallel training method. The scores in the Extractive column are the scores for the extractive summarizer trained in parallel with the abstractive summarizers. The scores in the Abstractive column are the scores for the abstractive summarizer trained in parallel with the extractive summarizers.

Extractive Model	Abstractive Model	PubMed						arXiv					
		Extractive			Abstractive			Extractive			Abstractive		
		R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L
HiStruct+	DYLE	46.66	20.47	42.17	46.41	17.33	41.74	45.28	17.63	40.23	46.61	18.14	41.73
	FACTORSUM	46.66	20.48	42.21	47.74	20.55	43.97	45.30	17.66	40.26	49.54	20.49	44.98
	PageSum	46.69	20.49	42.22	48.96	21.56	44.90	45.32	17.65	40.26	49.95	21.23	44.92
	HierGNN	46.73	20.50	42.23	49.85	21.94	45.56	45.34	17.69	40.28	50.12	21.14	45.08
HEGEL	DYLE	47.22	21.06	42.26	46.45	17.36	41.76	46.46	18.26	40.97	46.64	18.18	41.75
	FACTORSUM	47.23	21.13	42.30	47.79	20.58	44.01	46.55	18.29	41.02	49.57	20.51	45.01
	PageSum	47.28	21.16	42.32	49.01	21.59	44.94	46.55	18.32	41.02	50.01	21.26	44.97
	HierGNN	47.33	21.19	42.38	49.89	22.01	45.59	46.59	18.34	41.06	50.18	21.19	45.12
GRETEL	DYLE	48.28	21.27	43.13	46.47	17.38	41.80	48.27	20.40	42.69	46.61	18.20	41.79
	FACTORSUM	48.32	21.30	43.16	47.78	20.60	44.02	48.31	20.43	42.72	49.59	20.55	45.02
	PageSum	48.37	21.34	43.25	49.03	21.62	44.95	48.34	20.49	42.79	50.01	21.27	44.97
	HierGNN	48.39	21.40	43.28	49.93	22.01	45.62	48.39	20.54	42.80	50.20	21.23	45.14
Lodoss	DYLE	49.52	24.01	44.98	46.50	17.43	41.84	48.58	20.83	42.92	46.69	18.24	41.81
	FACTORSUM	49.57	24.06	45.00	47.85	20.66	44.08	48.66	20.89	42.97	49.63	20.90	45.07
	PageSum	49.61	24.13	45.05	49.07	21.67	45.02	48.69	20.97	42.98	50.05	21.30	45.02
	HierGNN	49.64	24.16	45.11	49.98	22.08	45.66	48.69	20.98	43.05	50.24	21.24	45.16

Table 7.10: Rouge scores for the sixteen combinations of four recently state-of-the-art extractive and abstractive summarizers on the PubMed and arXiv corpora using the parallel training method and the extractive-reference and the abstractive-reference similarity loss function.

Extractive Model	Abstractive Model	PubMed						arXiv					
		Extractive			Abstractive			Extractive			Abstractive		
		R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L
HiStruct+	DYLE	46.98	20.76	42.46	47.06	17.91	42.32	45.62	17.97	40.53	47.24	18.75	42.46
	FACTORSUM	47.03	20.76	42.49	48.08	20.95	44.37	45.64	17.99	40.53	49.82	20.80	45.31
	PageSum	47.04	20.79	42.52	49.34	21.93	45.30	45.65	18.01	40.57	50.25	21.11	45.25
	HierGNN	47.10	20.83	42.59	50.27	22.34	45.95	45.70	18.06	40.62	50.36	21.29	45.37
HEGEL	DYLE	47.45	21.46	42.63	47.11	18.02	42.46	46.89	18.60	40.28	47.27	18.81	42.42
	FACTORSUM	47.49	21.44	42.65	48.26	21.09	44.53	46.90	18.63	40.31	50.12	21.07	45.34
	PageSum	47.51	21.49	42.68	49.41	21.99	45.35	46.94	18.64	40.31	50.39	21.66	45.39
	HierGNN	47.59	21.52	42.74	50.32	22.46	46.03	47.03	18.69	40.42	50.55	21.50	45.44
GRETEL	DYLE	48.65	21.61	43.56	47.12	18.15	42.67	48.62	21.64	43.08	47.26	18.81	42.48
	FACTORSUM	48.71	21.65	43.60	48.17	20.94	44.39	48.67	21.72	43.10	49.94	20.88	45.31
	PageSum	48.76	21.74	43.65	49.50	22.04	45.37	48.73	21.78	43.19	50.35	21.17	45.49
	HierGNN	48.84	21.78	43.76	50.38	22.46	46.01	48.75	21.80	43.21	50.44	21.31	45.57
Lodoss	DYLE	49.97	24.44	45.40	47.44	18.36	42.88	49.01	21.26	43.10	47.74	18.93	42.84
	FACTORSUM	50.04	24.52	45.49	48.47	21.32	44.75	49.05	21.33	43.19	50.25	21.15	44.78
	PageSum	50.05	24.56	45.51	49.62	22.21	45.57	49.14	21.39	43.22	50.58	21.89	45.60
	HierGNN	50.15	24.64	45.59	50.48	22.57	46.17	49.20	21.48	43.28	50.73	21.97	45.69

If we consider the performance of the abstractive summarizers, DYLE, the least performing among the four, gains 0.19, 0.23, 0.26, and 0.31 Rouge-L scores (averaged over the Extractive and Abstractive scores) when trained with the parallel training approach only (see Tables 7.8, 7.9) when coupled with HiStruct+, HEGEL, GRETEL, and Lodoss, respectively. In this sce-

Table 7.11: Rouge scores for the sixteen combinations of four recently state-of-the-art extractive and abstractive summarizers on the PubMed and arXiv corpora using the parallel training method and the extractive-reference, the abstractive-reference, and the extractive-abstractive similarity loss function.

Extractive Model	Abstractive Model	PubMed						arXiv					
		Extractive			Abstractive			Extractive			Abstractive		
		R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L
HiStruct+	DYLE	47.02	20.77	42.48	47.12	17.98	42.39	45.63	17.99	40.55	47.31	18.82	42.51
	FACTORSUM	47.05	20.79	42.52	48.19	21.02	44.45	45.67	18.02	40.58	49.91	20.87	44.39
	PageSum	47.08	20.82	42.58	49.42	22.01	45.39	45.69	18.05	40.61	50.33	21.18	45.31
	HierGNN	47.12	20.86	42.63	50.34	22.42	46.02	45.74	18.09	40.66	50.42	21.36	45.44
HEGEL	DYLE	47.49	21.48	42.66	47.19	18.09	42.53	46.91	18.64	40.33	47.35	18.90	42.51
	FACTORSUM	47.51	21.49	42.69	48.35	21.17	44.61	46.94	18.66	40.35	50.19	21.11	45.43
	PageSum	47.55	21.53	42.72	49.52	22.09	45.44	47.00	18.68	40.39	50.47	21.55	45.39
	HierGNN	47.61	21.56	42.76	50.40	22.53	46.10	47.05	18.72	40.46	50.63	21.72	45.50
GRETEL	DYLE	48.69	21.63	43.60	47.18	18.22	42.74	48.66	21.67	43.12	47.38	18.91	42.59
	FACTORSUM	48.75	21.71	43.66	48.24	21.02	44.48	48.71	21.75	43.16	50.03	20.96	45.48
	PageSum	48.81	21.77	43.70	49.56	22.09	45.43	48.77	21.83	43.21	50.44	21.26	45.51
	HierGNN	48.86	21.82	43.78	50.45	22.49	46.08	48.79	21.84	43.24	50.51	21.39	45.54
Lodoss	DYLE	49.99	24.48	45.45	47.52	18.41	42.95	49.03	21.30	43.13	47.84	19.01	42.91
	FACTORSUM	50.04	24.56	45.51	48.58	21.41	44.85	49.09	21.37	43.21	50.37	21.22	45.53
	PageSum	50.08	24.60	45.54	49.71	22.29	45.65	49.16	21.43	43.26	50.68	21.96	45.67
	HierGNN	50.16	24.66	45.61	50.54	22.63	46.24	49.22	21.51	43.31	50.81	21.99	45.76

nario, for the best performing abstractive summarizer (among the considered ones), HierGNN, the performance boosts are 0.22, 0.24, 0.3, and 0.31. When the full semantic similarity loss function is used, the performance boosts observed for HierGNN are 0.67, 0.76, 0.81, and 1.1 when coupled with the extractive summarizers in the same order. This time the Rouge score boost found for the DYLE are 0.85, 0.92, 0.96, 1.4. Very similar patterns of results are observed for the other abstractive summarizers, as well. These values show that parallel training of the models improves the quality of the summarizers. But, when it is incorporated with the proposed semantic similarity loss function the models get another very noticeable performance boost.

Looking at the results and performance boosts of the extractive summarizers we make the following conclusions. Firstly, the semantic similarity-induced training method is both model-agnostic and corpus-agnostic. And for all of the combinations the performance boost is strictly monotonic. Furthermore, the improvement of any summarizer unit also depends on the quality of its counterpart. From Tables 7.9, 7.10, and 7.11, it is evident that the extractive summarizer achieves higher performance enhancements when combined with a more effective abstractive summarizer, and vice versa. In this experimental setting, the training process involves utilizing the top-performing extractive summarizer (Lodoss) in conjunction with the top-performing abstractive summarizer (HierGNN), resulting in the maximum performance boost and the highest Rouge values (note that PageSum is the top-performing abstractive summarizer for R-2 on the arXiv corpus).

The aforementioned findings suggest that even though employing the semantic similarity-induced parallel summarizer training approach enhances the performance of the summarizer pairs, selecting a more proficient counterpart will attain higher Rouge scores. Additionally, our observations indicate that abstractive summarizers benefit more from this training approach in terms of performance improvement compared to the extractive summarizers. One interesting observation that has emerged from our experiments is that our final results (Table 7.11 are counter to the findings reported in other research papers (Tables 7.7 and 7.8), that extractive summarizers exhibit superior performance on the PubMed corpus when compared to abstractive summarizers. Except for DYLE and a few other situations, Abstractive summarizers outperform Extractive summarizers on R-1 and R-L metrics. This remains the case for the arXiv corpus except for a few cases, most of which are in the R-2 metric.

7.3.5 Conclusions

This paper introduces a novel approach to training extractive and abstractive summarizers called semantic similarity-induced training method. Through parallel training and a modified loss function, both extractive and abstractive summarizers benefit from mutual guidance. Moreover, the inclusion of semantic distance in the loss function facilitates a closer alignment between the generated summaries and the reference summary in terms of semantic content. Future work will extend the study to test the conjecture that the parallel training method and new loss function will improve any combination of quality extractive and abstractive summarizers on any long document corpus.

7.3.6 Limitations

Despite the performance boost shown by the proposed training approach, it still has some limitations. Firstly, this training approach is tested for the long document only. That is why we are not sure how well it may perform for shorter texts. Secondly, Abstractive summarizers often suffer from hallucination [47]. This training approach may or may not overcome this incident of hallucination. Lastly, we are not sure whether this model is language specific or not. The performance of the summarizers may vary while summarizing text from different languages.

7.4 Conclusion

One novelty for the summarization task that we have introduced is the incorporation of the full citation network. To add background information, An et al. [10] utilized the referenced part

of the citation network, ignoring the citing part of the citation network. Yasunaga et al. [251] utilized the citing part of the citation network to reflect the impact of the considered article on the corresponding research community, but ignored the referenced part of the citation network. Our work is the first attempt to utilize both the citing and reference parts of the citation network. To accomplish this, we have introduced a scientific article summarization corpus comprising 10,000 research articles.

Like Cho et al. [47], we have incorporated the automatic segmentation as a part of the scientific article summarization task. In addition to this idea, we have incorporated the abstracts of the referenced articles as background information for each segment and applied an hierarchical summarization approach which has shown superior performance over the previous state-of-the-art (SOTA) models. Furthermore, our proposed models have the ability to generate both the extractive and abstractive summaries in parallel. The experimental results have shown that the parallel training of the extractive and abstractive summarizer units help both counterparts to attain significant performance boosts.

Following this finding we have introduced a novel training mechanism and a semantic-induced loss function for scientific document summarization. This training approach trains extractive and abstractive summarizers in parallel and the introduced loss function tries to bring the generated summaries closer to the reference summaries in the semantic space. The results have shown that this training approach significantly improves the performance of the SOTA extractive and abstractive summarizers; however, the abstractive summarizers attain higher performance gains.

Finally, to compare the performance of the summarizer model discussed in Section 7.2 to ChatGPT, we consider here, one sample (article ID: 0) from the arXiv scientific article summarization corpus [50]. The reference and the ChatGPT generated summaries are shown in Table 7.12. The ROUGE-1, ROUGE-2, ROUGE-L scores are 0.47, 0.16, and 0.27, respectively. These scores are lower than the abstractive summarizer model introduced in Section 7.2 even without using the citation network as the background information provider. This model has achieved 0.51 ROUGE-1, 0.22 ROUGE-2, and 0.46 ROUGE-L scores for the same sample.

The ChatGPT summary has some errors. Some are discussed here. The summary suggests that the study in the paper found that in 2011 millions of children in developing countries were malnourished. This information was from an analysis that the paper referred to, not the study reported in the paper. Also, “millions” suggests “a few million”, but the figure was 314 million. The prevalence of underweight, overweight, and obesity referred to high school girls not school-aged children. The improvements were in knowledge, not nutritional knowledge as mentioned in the summary, and the improvements were for girls and boys, not particularly for girls as mentioned in the summary.

Table 7.12: The reference summary and the ChatGPT generated summary for a sample from the arXiv corpus.

Reference Summary	Generated Summary
the present study was carried out to assess the effects of community nutrition intervention based on advocacy approach on malnutrition status among school-aged children in shiraz, iran. materials and methods: this case-control nutritional intervention has been done between 2008 and 2009 on 2897 primary and secondary school boys and girls (7-13 years old) based on advocacy approach in shiraz, iran. the project provided nutritious snacks in public schools over a 2-year period along with advocacy oriented actions in order to implement and promote nutritional intervention. for evaluation of effectiveness of the intervention growth monitoring indices of pre- and post-intervention were statistically compared. results: the frequency of subjects with body mass index lower than 5% decreased significantly after intervention among girls (p = 0.02). however, there were no significant changes among boys or total population. the mean of all anthropometric indices changed significantly after intervention both among girls and boys as well as in total population. the pre- and post-test education assessment in both groups showed that the student's average knowledge score has been significantly increased from 12.5 ± 3.2 to 16.8 ± 4.3 (p 0.0001). conclusion: this study demonstrates the potential success and scalability of school feeding programs in iran. community nutrition intervention based on the advocacy process model is effective on reducing the prevalence of underweight specifically among female school-aged children.	the text discusses a study on the prevalence of malnutrition among school-aged children in iran, focusing on the effectiveness of a community nutrition intervention based on an advocacy approach. the study found that in 2011, millions of children under five years old were stunted or underweight in developing countries. in iran, studies showed varying prevalence rates of underweight, overweight, and obesity among school-aged children. the text describes the implementation of a national free food program in elementary schools in deprived areas to cover poor students, but notes that many malnourished children in big cities are not covered by the program. the study aimed to improve the program by advocating for changes, such as providing nutritious and diverse snack packages for all children in need, including those in big cities. the intervention involved educational campaigns, providing nutritious snacks, and networking with stakeholders. the study found significant improvements in nutritional knowledge and anthropometric measures among the intervention group, particularly among girls. overall, the community nutrition intervention based on the advocacy process model was effective in reducing the prevalence of underweight, especially among female school-aged children.

Chapter 8

Conclusions

This thesis introduces a series of state-of-the-art deep learning architectures designed to tackle a spectrum of classical and intricate Natural Language Processing (NLP) challenges. Each of these works is encapsulated in its own chapter, ensuring a self-contained and comprehensive coverage. Exhibiting a robust theoretical foundation, rigorous investigation, and extensive experimentation, all presented works have been featured in or prepared for respected conferences, attesting to their academic significance. Readers are guided through a journey that unveils the intuitive thinking process behind addressing various NLP problems. The consistent thread running through this thesis is the integration of cutting-edge deep learning models with foundational knowledge in natural language. In this chapter, we distill our key findings, outline major contributions, and acknowledge the limitations inherent in this thesis. Moreover, we delineate future directions for research, outlining potential areas for improvement and expansion. This synthesis serves as both a culmination of the presented works and a springboard for future exploration in the dynamic realm of natural language processing.

8.1 Key Findings

This study presents a comprehensive research landscape, engaging the reader in a multifaceted exploration of NLP problems. Delving into the intricacies of semantic similarity measurement, relation extraction, document classification, and text summarization, we have addressed each of these challenges with tailored models. The citation linkage problem is conceptualized as a semantic similarity measurement task, while protein-protein (PPI) and drug-drug (DDI) interaction identification are approached as relation extraction tasks. Personality trait identification is tackled through two distinct methodologies: semantic similarity measurement and document classification. In the realm of scientific text summarization, we have delved into both extractive and abstractive summarization approaches, offering a comprehensive investigation into diverse

facets of NLP.

Our objective was to incorporate syntactic features and enhance semantic preservation in downstream NLP applications. We observed a prevalent tendency in state-of-the-art models to overlook the structural and grammatical aspects of textual representations within these domains. To address this research gap, we have investigated the integration of constituency and dependency tree information in tasks related to citation linkage, Protein-Protein Interaction (PPI), Drug-Drug Interaction (DDI), and personality trait identification. By leveraging dependency and constituency tree transformers, our investigated models effectively retained phrasal and inter-word dependency information. This integration played a pivotal role in facilitating performance enhancements. The utilization of these additional structural and grammatical features resulted in our proposed models achieving state-of-the-art performances across various scenarios.

Furthermore, to enhance the semantic preservation of the models, we have introduced a word-refinement module designed to enhance word embeddings with context information. Across tasks involving PPI, DDI, personality trait analysis, and summarization, our models have demonstrated state-of-the-art performances by leveraging context-aware word representations. While large language models offer context-aware word representations, their task-specific fine-tuning demands substantial computational resources. In contrast, our proposed models utilize a graph attention network to generate enriched context-aware word embeddings, thereby achieving superior performance with reduced computational resource demands.

A pivotal discovery in our research is the efficacy of multitask training in enhancing the performance of summarizer models. In our experiments, we adopted a joint training approach for extractive and abstractive summarizers, alongside the incorporation of a segmentation model. This innovative strategy, coupled with the utilization of citation network information, empowered our investigated summarizers to integrate background information on a segment-by-segment basis. This stands in contrast to other models that assimilate all background information as a single unit. The proposed model, leveraging multitask training and segment-wise background information integration, outperformed alternative models, showcasing superior performances across various metrics.

8.2 Major Contributions

This study embarks on an extensive exploration of diverse NLP problems, employing innovative deep learning models characterized by intuitive architectures. The elucidation of the modules employed within these architectures is presented with clarity, providing an intuitive understanding of their selection and role. Beyond the intuitive design principles, the study at-

tains state-of-the-art performance on several tasks, marking a significant advancement at the time of model publication. The culmination of these endeavors results in a set of major contributions, summarized as follows:

- We have pioneered a novel methodology for the creation of synthetic corpora, specifically tailored for semantic similarity tasks. This innovative approach has culminated in the development of a corpus comprising 74,568 samples designed for the citation linkage task within the biomedical research domain.

- Our implementation of models for tasks involving semantic similarity measurement, relation extraction, and document classification is distinguished by the incorporation of syntactic information. Notably, our models have achieved state-of-the-art performances in these domains, underscoring their efficacy and advancement beyond existing benchmarks.

- Our models have surmounted the token intake limit inherent in traditional BERT-based models. This achievement was realized by leveraging tree-structured neural networks as sentence encoders, complemented by a graph attention network serving as an overarching mechanism to interconnect them. This innovative approach allows our models to handle texts of variable lengths, transcending the constraints posed by the token intake limits of conventional BERT-based architectures.

- The introduction of our word-refinement module represents a breakthrough, enabling downstream tasks to benefit from task-specific, context-aware word embeddings. This methodology aligns with the objectives of BERT fine-tuning but does so with reduced computational resource requirements.

- We have introduced a semantic-induced joint training approach for both extractive and abstractive summarizers. This innovative methodology significantly enhances the performance of each individual summarizer, marking a notable advancement in summarization techniques.

- To the best of our knowledge, we stand as trailblazers in the utilization of both sides of the citation network—citing and cited—for scientific document summarization. Our scientific document summarizers exhibit a unique capability, going beyond the provision of the core content of research articles. Our scientific document summarizers go beyond merely offering the essence of research articles; they possess the unique capability of reflecting the impact of specific research within the corresponding research community over time, achieved through a thorough analysis of citing statements. Furthermore, our

summarizers acquire essential background knowledge by capturing the abstracts of the reference articles.

- Our summarizer models possess the capability to incorporate background information segment-wise through the joint training of a neural segmentation model. This innovative approach ensures that the summarizer model receives information in a segmented and organized manner, facilitating a proper flow of information. This strategic integration has proven instrumental in achieving a substantial performance improvement compared to existing prominent works in the field.

- Leveraging the citation network, we have introduced a pioneering corpus for scientific document summarization encompassing 10,000 research articles. This corpus not only links to the articles referenced in the considered documents but also incorporates citations made to them.

8.3 Limitations of the Study

In this thesis, we examine every facet of an architecture, conduct a thorough analysis of the advantages and disadvantages of each linguistic feature incorporated, and consistently achieve commendable results. However, it is crucial to acknowledge that there are still a few limitations that warrant consideration. Chapter 5 Section 5.3, Chapter 6 Section 6.3, and Chapter 7 Sections 7.1, 7.2, and 7.3 individually delineate their inherent limitations, while the remaining chapters articulate their constraints within the purview of result analysis and conclusive discussions. Nevertheless, a comprehensive overview of the limitations intrinsic to all scrutinized models is presented herein.

The citation linkage framework has demonstrated promising performance on both the created silver and gold-standard corpora [85]. Nevertheless, a notable limitation of the introduced framework lies in its confinement to the sentence-level, which, upon reviewing scientific documents, appears restrictive. We have observed a need to broaden its scope to the paragraph level, recognizing the inherent interconnectedness of information. An initiative was taken to create a gold-standard corpus for this expanded task, aiming to encompass a broader span beyond sentence-level similarity. However, due to the resource-intensive nature of this task, particularly in terms of time and the necessity for annotators with specialized knowledge, the endeavour had to be abandoned.

Similarly, the relation extraction task between biomedical entities from research articles is encapsulated at the sentence-level. The relation between different biomedical entities may be found after analysing multiple sentences rather than considering a single sentence only.

Regrettably, akin to the previously mentioned case, there is no corpus available which expands its scope beyond single sentences. Consequently, our models rely solely on single sentences to identify relationships between different biomedical entities.

In addressing the personality trait identification task, our primary emphasis was placed on the Big Five Model (OCEAN) and the Myers-Briggs Type Indicator (MBTI) classifications. Nonetheless, it is crucial to acknowledge the presence of two additional notable personality trait models—Eysenck's Personality Dimensions and the HEXACO Model. These alternative frameworks provide unique perspectives for comprehending and classifying personality traits. The performance and adaptability of our models to these alternative models remain uncertain.

While the models incorporating word-refinement modules have demonstrated a significant boost in performance, it is imperative to recognize a trade-off in terms of computational time. In the initial forward pass, the model utilizes RoBERTa word embeddings to generate sentence and statement representations, updating word representations from this generated information. In the subsequent forward pass, using the context-enriched word embeddings, the model regenerates sentence and statement representations for downstream tasks. The use of two forward passes coupled with the parsing required for the tree-structured transformers in the model contributes to an increased time requirement for result generation compared to other models. This computational overhead should be duly considered when contemplating the deployment and scalability of the proposed models in practical applications.

The joint training of extractive and abstractive summarizer units involved extensive experimentation with a substantial number of epochs. To safeguard against overfitting, we diligently monitored the validation loss curve after every 5 epochs. This process, while effective, is computationally expensive and necessitates an extended period for model training.

8.4 Recommendations for Future Research

Throughout Chapters 4 to 7, certain sections propose future research directions pertaining to specific topics. While some of these recommended avenues have been subsequently addressed in later chapters as the thesis unfolded, others remain as unexplored possibilities awaiting investigation in future research endeavours.

To enhance the applicability of the initial two tasks, citation linkage and relation extraction between biomedical entities, a significant avenue for improvement involves extending their application range beyond single sentences. This could be achieved by creating corpora that expand from the sentence level to the paragraph level for these two tasks, and to the document level for the latter task. The design of models aligned with these expanded applications could render them more useful and practical in real-life scenarios.

In the context of relation extraction tasks, a noteworthy avenue for improvement involves the incorporation of task-specific knowledge graphs. A notable example is the work by Asada et al. [15], where they integrated the knowledge graph of drugs for Drug-Drug Interaction (DDI). While our models have demonstrated significant performance enhancements without the use of additional task-specific features, integrating such features could further elevate their performance by providing enhanced reasoning capabilities.

For the personality trait identification, exploring the integration of Language Inquiry and Word Count (LIWC) between posts from social media could yield valuable insights. Additionally, leveraging knowledge graphs by incorporating interlinked descriptions of concepts, entities, and relationships in a machine-readable form, as introduced by Ramezani et al. [179], stands as a potential avenue for enhancement. These approaches may contribute to a richer understanding of personality traits and improve the models' performance in this domain.

In the context of scientific document summarization, while we have successfully incorporated the abstracts of reference articles as background information, there is an opportunity to further enrich this background information by utilizing the citation linkage framework. This entails fetching sentences from reference articles that are semantically similar to the referencing statements and using them alongside the abstracts as the background information. As discussed in Chapter 7 Section 7.2, the impact of a research work may evolve over time, and such changes in impact may not be adequately reflected in the abstract alone. Therefore, relying solely on abstracts may not suffice to provide the necessary background information for summarizing the considered article. Initial experiments from our side on this idea have shown promising results, but further exploration and experimentation are required to solidify these claims and present them as established facts.

Moreover, the integration of tree-structured neural networks for sentence encoding in our summarization models holds the potential to enhance model performance by incorporating additional syntactic information. This avenue presents an opportunity to further investigate the impact of syntactic structures on summarization quality and explore potential improvements in capturing nuanced relationships within textual content.

Bibliography

[1] Hervé Abdi and Lynne J Williams. Principal component analysis. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(4):433–459, 2010.

[2] Ahmed AbuRa'ed, Luis Chiruzzo, and Horacio Saggion. What sentence are you referring to and why? Identifying cited sentences in scientific literature. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, pages 9–17, 2017.

[3] Joel Achenbach. Coronavirus is harming the mental health of tens of millions of people in U.S., new poll finds. `https://www.washingtonpost.com/health/coronavirus-is-harming-the-mental-health-of-tens-of-millions-of-people-in-us-new-poll-finds/2020/04/02/565e6744-74ee-11ea-85cb-8670579b863d_story.html`, 2020.

[4] Mahtab Ahmed, Jumayel Islam, Muhammad Rifayat Samee, and Robert E Mercer. Identifying protein-protein interaction using tree LSTM and structured attention. In *2019 IEEE 13th International Conference on Semantic Computing (ICSC)*, pages 224–231, 2019.

[5] Mahtab Ahmed, Muhammad Rifayat Samee, and Robert E Mercer. Improving tree-LSTM with tree attention. In *2019 IEEE 13th International Conference on Semantic Computing (ICSC)*, pages 247–254, 2019.

[6] Mahtab Ahmed, Muhammad Rifayat Samee, and Robert E Mercer. You only need attention to traverse trees. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 316–322, 2019.

[7] Antti Airola, Sampo Pyysalo, Jari Björne, Tapio Pahikkala, Filip Ginter, and Tapio Salakoski. All-paths graph kernel for protein-protein interaction extraction with evaluation of cross-corpus learning. *BMC Bioinformatics*, 9 Suppl 11:S2, 11 2008.

[8] Dmitrii Aksenov, Julian Moreno Schneider, Peter Bourgonje, Robert Schwarzenberg, Leonhard Hennig, and Georg Rehm. Abstractive text summarization based on language model conditioning and locality modeling. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 6680–6689, 2020.

[9] Melina Altmann, Stefan Altmann, Patricia A Rodriguez, Benjamin Weller, Lena Elorduy Vergara, Julius Palme, Nora Marín-de la Rosa, Mayra Sauer, Marion Wenig, José Antonio Villaécija-Aguilar, et al. Extensive signal integration by the phytohormone protein network. *Nature*, 583(7815):271–276, 2020.

[10] Chenxin An, Ming Zhong, Yiran Chen, Danqing Wang, Xipeng Qiu, and Xuanjing Huang. Enhancing scientific papers summarization with citation graph. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 12498–12506, 2021.

[11] Rahul Aralikatte, Shashi Narayan, Joshua Maynez, Sascha Rothe, and Ryan McDonald. Focus attention: Promoting faithfulness and diversity in summarization. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6078–6095, 2021.

[12] Shlomo Argamon, Sushant Dhawle, Moshe Koppel, and James W Pennebaker. Lexical predictors of personality type. In *Proceedings of the 2005 Joint Annual Meeting of the Interface and the Classification Society of North America*, pages 1–16, 2005.

[13] Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador García, Sergio Gil-López, Daniel Molina, Richard Benjamins, et al. Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion*, 58:82–115, 2020.

[14] Masaki Asada, Makoto Miwa, and Yutaka Sasaki. Using drug descriptions and molecular structures for drug–drug interaction extraction from literature. *Bioinformatics*, 37(12):1739–1746, 2021.

[15] Masaki Asada, Makoto Miwa, and Yutaka Sasaki. Integrating heterogeneous knowledge graphs into drug–drug interaction extraction from the literature. *Bioinformatics*, 39(1):btac754, 2023.

[16] Awais Athar. Sentiment analysis of citations using sentence structure-based features. In *Proceedings of the ACL 2011 Student Session*, pages 81–87, 2011.

[17] Awais Athar. Sentiment analysis of scientific citations. Technical report, Computer Laboratory, University of Cambridge, 2014.

[18] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, Conference Track Proceedings*, 2014.

[19] Gaurav Baruah and Maheedhar Kolla. Klick labs at cl-scisumm 2018. In *BIRNDL@ SIGIR*, 2018.

[20] William A Baumgartner, Zhiyong Lu, Helen L Johnson, J Gregory Caporaso, Jesse Paquette, Anna Lindemann, Elizabeth K White, Olga Medvedeva, K Bretonnel Cohen, and Lawrence Hunter. Concept recognition for extracting protein interaction relations from biomedical text. *Genome Biology*, 9(2):1–15, 2008.

[21] Iz Beltagy, Kyle Lo, and Arman Cohan. SciBERT: A pretrained language model for scientific text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3615–3620, 2019.

[22] Iz Beltagy, Matthew E Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020.

[23] Verónica Benet-Martínez and Oliver P John. Los cinco grandes across cultures and ethnic groups: Multitrait-multimethod analyses of the big five in spanish and english. *Journal of Personality and Social Psychology*, 75(3):729, 1998.

[24] Shane Bergsma. Automatic acquisition of gender information for anaphora resolution. In *Proceedings of the 18th Conference of the Canadian Society for Computational Studies of Intelligence*, pages 342–353, 2005.

[25] Shane Bergsma and Dekang Lin. Bootstrapping path-based pronoun resolution. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 33–40, 2006.

[26] Shane Bergsma and Benjamin Van Durme. Using conceptual class attributes to characterize social media users. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 710–720, 2013.

[27] Christian Blaschke, Miguel A Andrade, Christos A Ouzounis, and Alfonso Valencia. Automatic extraction of biological information from scientific text: protein-protein interactions. In *Seventh International Conference on Intelligent Systems for Molecular Biology*, pages 60–67, 1999.

[28] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017.

[29] Serena Bonin, F Petrera, B Niccolini, and Giorgio Stanta. PCR analysis in archival postmortem tissues. *Molecular Pathology*, 56(3):184–186, 2003.

[30] Sabri Boughorbel, Fethi Jarray, and Mohammed El-Anbari. Optimal classifier for imbalanced data using Matthews correlation coefficient metric. *PloS ONE*, 12(6), 2017.

[31] Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, 2015.

[32] I Briggs Myers. *Introduction to Type: A Guide to Understanding Your Results on the Myers-Briggs Type Indicator (revised by LK Kirby & KD Myers)*. CA: Consulting Psychologists Press, 1993.

[33] Kay Henning Brodersen, Cheng Soon Ong, Klaas Enno Stephan, and Joachim M Buhmann. The balanced accuracy and its posterior distribution. In *Proceedings of the 2010 20th International Conference on Pattern Recognition*, pages 3121–3124, 2010.

[34] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, pages 1877–1901, 2020.

[35] Razvan Bunescu, Ruifang Ge, Rohit Kate, Edward Marcotte, Raymond Mooney, Arun Ramani, and Yuk Wah Wong. Comparative experiments on learning information extrac-

tors for proteins and their interactions. *Artificial Intelligence in Medicine*, 33(2):139–155, 2005.

[36] Isabel Cachola, Kyle Lo, Arman Cohan, and Daniel S Weld. TLDR: Extreme summarization of scientific documents. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4766–4777, 2020.

[37] Erik Cambria, Soujanya Poria, Alexander Gelbukh, and Mike Thelwall. Sentiment analysis is a big suitcase. *IEEE Intelligent Systems*, 32(6):74–80, 2017.

[38] Laura Campbell-Sills, Sharon L Cohan, and Murray B Stein. Relationship of resilience to personality, coping, and psychiatric symptoms in young adults. *Behaviour Research and Therapy*, 44(4):585–599, 2006.

[39] Ziqiang Cao, Wenjie Li, and Dapeng Wu. Polyu at cl-scisumm 2016. In *Proceedings of the Joint Workshop on Bibliometric-enhanced Information Retrieval and Natural Language Processing for Digital Libraries (BIRNDL)*, pages 132–138, 2016.

[40] Fabio Celli, Bruno Lepri, Joan-Isaac Biel, Daniel Gatica-Perez, Giuseppe Riccardi, and Fabio Pianesi. The workshop on computational personality recognition 2014. In *Proceedings of the 22nd ACM International Conference on Multimedia*, pages 1245–1246, 2014.

[41] Muthu Kumar Chandrasekaran, Guy Feigenblat, Eduard Hovy, Abhilasha Ravichander, Michal Shmueli-Scheuer, and Anita De Waard. Overview and insights from scientific document summarization shared tasks 2020: Cl-SciSumm, LaySumm and LongSumm. In *Proceedings of the First Workshop on Scholarly Document Processing (SDP 2020)*, 2020.

[42] Yung-Chun Chang, Chun-Han Chu, Yu-Chen Su, Chien Chin Chen, and Wen-Lian Hsu. Pipe: a protein–protein interaction passage extraction module for BioCreative challenge. *Database*, 2016, 2016.

[43] Jagat S Chauhan, Nitish K Mishra, and Gajendra PS Raghava. Identification of atp binding residues of a protein from its primary sequence. *BMC Bioinformatics*, 10(1):1–9, 2009.

[44] Danqi Chen and Christopher D Manning. A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in NaturalLanguage Processing (EMNLP)*, pages 740–750, 2014.

[45] Jianpeng Cheng and Maria Lapata. Neural summarization by extracting sentences and words. In *54th Annual Meeting of the Association for Computational Linguistics*, pages 484–494, 2016.

[46] Luis Chiruzzo, Ahmed AbuRa'ed, Àlex Bravo, Horacio Saggion, et al. LaSTUS-TALN+INCO @ CL-SciSumm 2019. In *CEUR Workshop Proceedings*, volume 1181, 2019.

[47] Sangwoo Cho, Kaiqiang Song, Xiaoyang Wang, Fei Liu, and Dong Yu. Toward unifying text segmentation and long document summarization. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 106–118, 2022.

[48] Sangwoo Cho, Kaiqiang Song, Xiaoyang Wang, Fei Liu, and Dong Yu. Toward unifying text segmentation and long document summarization. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 106–118, 2022.

[49] Sung-Pil Choi. Extraction of protein-protein interactions (ppis) from the literature by deep convolutional neural networks with various feature embeddings. *Journal of Information Science*, 44, 11 2016.

[50] Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. A discourse-aware attention model for abstractive summarization of long documents. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 615–621, 2018.

[51] Arman Cohan and Nazli Goharian. Scientific document summarization via citation contextualization and scientific discourse. *International Journal on Digital Libraries*, 19(2):287–303, 2018.

[52] Jacob Cohen. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46, 1960.

[53] Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680, 2017.

[54] Paul T Costa Jr and Robert R McCrae. Domains and facets: Hierarchical personality assessment using the revised neo personality inventory. *Journal of Personality Assessment*, 64(1):21–50, 1995.

[55] Harald Cramér. *Mathematical Methods of Statistics*, volume 9 of *Mathematical Methods of Statistics (PMS-9)*. Princeton University Press, 1946.

[56] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, 2019.

[57] Jing Ding, Daniel Berleant, Dan Nettleton, and Eve Wurtele. Mining medline: abstracts, sentences, or phrases? In *Pacific Symposium on Biocomputing 7*, pages 326–337, 2001.

[58] Filip Karlo Došilović, Mario Brčić, and Nikica Hlupić. Explainable artificial intelligence: A survey. In *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pages 0210–0215, 2018.

[59] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(7), 2011.

[60] Gunes Erkan, Arzucan Özgür, and Dragomir Radev. Semi-supervised classification for extracting protein interaction sentences using dependency parsing. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 228–237, 2007.

[61] Günes Erkan and Dragomir R Radev. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research*, 22:457–479, 2004.

[62] Yang Fan, Shufang Xie, Yingce Xia, Lijun Wu, Tao Qin, Xiang-Yang Li, and Tie-Yan Liu. Multi-branch attentive transformer. *arXiv preprint arXiv:2006.10270*, 2020.

[63] Golnoosh Farnadi, Susana Zoghbi, Marie-Francine Moens, and Martine De Cock. How well do your facebook status updates express your personality? In *Proceedings of the 22nd Edition of the Annual Belgian-Dutch Conference on Machine Learning (BENELEARN)*, page 88, 2013.

[64] Hao Fei, Yue Zhang, Yafeng Ren, and Donghong Ji. A span-graph neural model for overlapping entity relation extraction in biomedical texts. *Bioinformatics*, 37(11):1581–1589, 2021.

[65] Ahmad Firoz, Adeel Malik, Karl H Joplin, Zulfiqar Ahmad, Vivekanand Jha, and Shandar Ahmad. Residue propensities, discrimination and binding site prediction of adenine and guanine phosphates. *BMC Biochemistry*, 12(1):1–12, 2011.

[66] Marcio Fonseca, Yftah Ziser, and Shay B. Cohen. Factorizing content and budget decisions in abstractive summarization of long documents. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6341–6364, 2022.

[67] Alyssa Fowers and William Wan. A third of Americans now show signs of clinical anxiety or depression, Census Bureau finds amid coronavirus pandemic. https://www.washingtonpost.com/health/2020/05/26/americans-with-depression-anxiety-pandemic/?arc404=true, 2020.

[68] Xiyan Fu, Jun Wang, Jinghan Zhang, Jinmao Wei, and Zhenglu Yang. Document summarization with vhtm: Variational hierarchical topic-aware mechanism. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7740–7747, 2020.

[69] Katrin Fundel, Robert Küffner, and Ralf Zimmer. RelEx—Relation extraction using dependency parse trees. *Bioinformatics*, 23(3):365–371, 2007.

[70] Eugene Garfield. Citation analysis as a tool in journal evaluation. *Science*, 178(4060):471–479, 1972.

[71] Mark Garzone and Robert E Mercer. Towards an automated citation classifier. In *Proceedings of the 13th Biennial Conference of the Canadian Society for Computational Studies of Intelligence*, pages 337–346, 2000.

[72] Alexios Gidiotis, Stefanos Stefanidis, and Grigorios Tsoumakas. AUTH @ CLSciSumm 20, LaySumm 20, LongSumm 20. In *Proceedings of the First Workshop on Scholarly Document Processing*, pages 251–260, 2020.

[73] Alexios Gidiotis and Grigorios Tsoumakas. A divide-and-conquer approach to the summarization of long documents. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:3029–3040, 2020.

[74] Matej Gjurković, Mladen Karan, Iva Vukojević, Mihaela Bošnjak, and Jan Snajder. PANDORA talks: Personality and demographics on Reddit. In *Proceedings of the Ninth International Workshop on Natural Language Processing for Social Media*, pages 138–152, 2021.

[75] David P Goldberg and Peter Huxley. *Common Mental Disorders: A Bio-social Model*. Tavistock/Routledge, 1992.

[76] Lewis R Goldberg. International personality item pool: A scientific collaboratory for the development of advanced measures of personality and other individual differences. *Retrieved August*, 14:2002, 1999.

[77] David E Gordon, Gwendolyn M Jang, Mehdi Bouhaddou, Jiewei Xu, Kirsten Obernier, Kris M White, Matthew J O'Meara, Veronica V Rezelj, Jeffrey Z Guo, Danielle L Swaney, et al. A SARS-CoV-2 protein interaction map reveals targets for drug repurposing. *Nature*, 583(7816):459–468, 2020.

[78] Hannes Grassegger and Mikael Krogerus. The data that turned the world upside down. *Vice Motherboard*, 28, 2017.

[79] Robert J. Gregory. The history of psychological testing. In *Psychological Testing: History, principles, and applications.*, chapter 2, pages 32–58. Pearson, 7th edition, 2013.

[80] Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. Domain-specific language model pretraining for biomedical natural language processing. *ACM Transactions on Computing for Healthcare*, 3(1):1–23, 2021.

[81] Supriya Gupta, Aakanksha Sharaff, and Naresh Kumar Nagwani. Biomedical text summarization: a graph-based ranking approach. In *Applied Information Processing Systems*, volume 1354, pages 147–156. Springer, 2022.

[82] Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, 2020.

[83] Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. Teaching machines to read and comprehend.

In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, pages 1693–1701, 2015.

[84] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[85] Hospice Houngbo and Robert E Mercer. Investigating citation linkage with machine learning. In *Proceedings of the 30th Canadian Conference on Artificial Intelligence*, pages 78–83, 2017.

[86] Kokou Hospice Houngbo. *Investigating Citation Linkage Between Research Articles*. PhD thesis, The University of Western Ontario, 2017.

[87] OM Zack Howard, Hui Fang Dong, Aiko-Konno Shirakawa, and Joost J Oppenheim. LEC induces chemotaxis and adhesion by interacting with CCR1 and CCR8. *Blood, The Journal of the American Society of Hematology*, 96(3):840–845, 2000.

[88] Yu-Lun Hsieh, Yung-Chun Chang, Nai-Wen Chang, and Wen-Lian Hsu. Identifying protein-protein interactions in biomedical literature using recurrent neural networks with long short-term memory. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Vol. 2: Short Papers)*, pages 240–245, 2017.

[89] Lei Hua and Chanqin Quan. A shortest dependency path based convolutional neural network for protein-protein relation extraction. *BioMed Research International*, 2016, 2016.

[90] Dandan Huang, Leyang Cui, Sen Yang, Guangsheng Bao, Kun Wang, Jun Xie, and Yue Zhang. What have we achieved on text summarization? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 446–469, 2020.

[91] Cornelis JJ Huijsmans, Jan Damen, Johannes C van der Linden, Paul HM Savelkoul, and Mirjam HA Hermans. Comparative analysis of four methods to extract DNA from paraffin-embedded tissues: Effect on downstream molecular applications. *BMC Research Notes*, 3(1):239, 2010.

[92] Machine Learning Interviews. METEOR metric for machine translation. `https://machinelearninginterview.com/topics/machine-learning/meteor-for-machine-translation/`. Accessed: 2023-12-15.

[93] Kokil Jaidka, Muthu Kumar Chandrasekaran, Sajal Rustagi, and Min-Yen Kan. Overview of the CL-SciSumm 2016 shared task. In *Proceedings of the Joint Workshop on Bibliometric-enhanced Information Retrieval and Natural Language Processing for Digital Libraries (BIRNDL)*, pages 93–102, 2016.

[94] Kokil Jaidka, Michihiro Yasunaga, Muthu Kumar Chandrasekaran, Dragomir Radev, and Min-Yen Kan. The CL-SciSumm shared task 2018: Results and key insights. In *CEUR Proceedings*, volume 2132, 2019.

[95] O. P. John, L.P. Naumann, and C.J. & Soto. Paradigm shift to the integrative big five taxonomy: History, measurement, and conceptual issues. In *Handbook of Personality: Theory and Research*, pages 114–158. The Guilford Press, 2008.

[96] Oliver P John, Eileen M Donahue, and Robert L Kentle. Big five inventory. *Journal of Personality and Social Psychology*, 1991.

[97] Oliver P. John, Laura P. Naumann, and Cristopher J. Soto. Paradigm shift to the integrative Big Five Trait taxonomy. *Handbook of personality: Theory and research*, pages 114–158, 2008.

[98] Deborah G Johnson. *Computer Ethics*. Prentice Hall, 1985.

[99] Mitchell Jolly. (MBTI) Myers-Briggs personality type dataset. `https://www.kaggle.com/datasnaek/mbti-type`, 2017.

[100] Daniel Jurafsky and James H. Martin. *Speech and Language Processing*. New Jersey: Prentice-Hall, Inc, 2012.

[101] Mayuri Pundlik Kalghatgi, Manjula Ramannavar, and Nandini S Sidnal. A neural network approach to personality prediction based on the big-five model. *International Journal of Innovative Research in Advanced Engineering (IJIRAE)*, 2(8):56–63, 2015.

[102] Neel Kanwal and Giuseppe Rizzo. Attention-based clinical note summarization. In *Proceedings of the 37th ACM/SIGAPP Symposium on Applied Computing*, pages 813–820, 2022.

[103] Mahmut Kaya and Hasan Sakir Bilge. Deep metric learning: A survey. *Symmetry*, 11:1066, 2019.

[104] K Kayser, H Stute, J Lübcke, and U Wazinski. Rapid microwave fixation—a comparative morphometric study. *The Histochemical Journal*, 20(6-7):347–352, 1988.

[105] Amirmohammad Kazameini, Samin Fatehi, Yash Mehta, Sauleh Eetemadi, and Erik Cambria. Personality trait detection using bagged SVM over BERT word embedding ensembles. In *Proceedings of the The Fourth Widening Natural Language Processing Workshop*, 2020.

[106] Amirmohammad Kazemeini, Sudipta Singha Roy, Robert E Mercer, and Erik Cambria. Interpretable representation learning for personality detection. In *2021 International Conference on Data Mining Workshops (ICDMW)*, pages 158–165, 2021.

[107] Kenneth S Kendler, Ronald C Kessler, Michael C Neale, Andrew C Heath, and Lindon J Eaves. The prediction of major depression in women: toward an integrated etiologic model. *American Journal of Psychiatry*, 150:1139–1139, 1993.

[108] Jihyun Kim, Kelly Merrill Jr, Chad Collins, and Hocheol Yang. Social TV viewing during the COVID-19 lockdown: The mediating role of social presence. *Technology in Society*, 67:101733, 2021.

[109] Seonho Kim, Juntae Yoon, Jihoon Yang, and Seog Park. Walk-weighted subsequence kernels for protein-protein interaction extraction. *BMC Bioinformatics*, 11(1):1–21, 2010.

[110] Terry Koo, Amir Globerson, Xavier Carreras Pérez, and Michael Collins. Structured prediction models via the matrix-tree theorem. In *Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 141–150, 2007.

[111] Michal Kosinski, David Stillwell, and Thore Graepel. Private traits and attributes are predictable from digital records of human behavior. *Proceedings of the National Academy of Sciences*, 110(15):5802–5805, 2013.

[112] Martin Krallinger, Florian Leitner, Carlos Rodriguez-Penagos, and Alfonso Valencia. Overview of the protein-protein interaction annotation extraction task of BioCreative II. *Genome Biology*, 9(2):1–19, 2008.

[113] Souvik Kundu. Citation polarity identification from scientific articles using deep learning methods. Master's thesis, The University of Western Ontario, 2023.

[114] Artuur Leeuwenberg, Aleksey Buzmakov, Yannick Toussaint, and Amedeo Napoli. Exploring pattern structures of syntactic trees for relation extraction. In *International Conference on Formal Concept Analysis*, pages 153–168, 2015.

[115] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, 2020.

[116] Patrick Lewis, Myle Ott, Jingfei Du, and Veselin Stoyanov. Pretrained language models for biomedical and clinical tasks: Understanding and extending the state-of-the-art. In *Proceedings of the 3rd Clinical Natural Language Processing Workshop*, pages 146–157, 2020.

[117] Richard Lewis. *When Cultures Collide*. Nicholas Brealey Publishing Boston, MA, 2010.

[118] Jiwei Li, Minh-Thang Luong, Dan Jurafsky, and Eudard Hovy. When are tree structures necessary for deep learning of representations? In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2304–2314, 2015.

[119] Lei Li, Liyuan Mao, Yazhao Zhang, Junqi Chi, Taiwen Huang, Xiaoyue Cong, and Heng Peng. Computational linguistics literature and citations oriented citation linkage, classification and summarization. *International Journal on Digital Libraries*, 19(2-3):173–190, 2018.

[120] Lei Li, Yang Xie, Wei Liu, Yinan Liu, Yafei Jiang, Siya Qi, and Xingyuan Li. CIST @ CL-SciSumm 2020, LongSumm 2020: Automatic scientific document summarization. In *Proceedings of the First Workshop on Scholarly Document Processing*, pages 225–234, 2020.

[121] Lei Li, Yazhao Zhang, Liyuan Mao, Junqi Chi, Moye Chen, and Zuying Huang. CIST@CLSciSumm-17: Multiple features based citation linkage, classification and summarization. In *BIRNDL 2017*, pages 43–54, 2017.

[122] Lei Li, Yingqi Zhu, Yang Xie, Zuying Huang, Wei Liu, Xingyuan Li, and Yinan Liu. CIST@CLSciSumm-19: Automatic scientific paper summarization with citances and facets. In *BIRNDL 2019*, 2019.

[123] Cynthia CS Liem, Markus Langer, Andrew Demetriou, Annemarie MF Hiemstra, Achmadnoer Sukma Wicaksana, Marise Ph Born, and Cornelius J König. Psychology meets machine learning: Interdisciplinary perspectives on algorithmic job candidate screening. In *Explainable and Interpretable Models in Computer Vision and Machine Learning*, pages 197–253. Springer, 2018.

[124] Fei Liu, Julien Perez, and Scott Nowson. A language-independent and compositional model for personality trait recognition from short texts. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017, Volume 1: Long Papers*, pages 754–764, 2017.

[125] Shengyu Liu, Buzhou Tang, Qingcai Chen, and Xiaolong Wang. Drug-drug interaction extraction via convolutional neural networks. *Computational and Mathematical Methods in Medicine*, 2016, 2016.

[126] Yang Liu and Mirella Lapata. Text summarization with pretrained encoders. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3730–3740, 2019.

[127] Yang Liu, Chengjie Sun, Lei Lin, and Xiaolong Wang. Learning natural language inference using bidirectional LSTM model and inner-attention. *arXiv preprint arXiv:1605.09090*, 2016.

[128] Yang Liu, Ivan Titov, and Mirella Lapata. Single document summarization as tree induction. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1745–1755, 2019.

[129] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[130] Yixin Liu, Ansong Ni, Linyong Nan, Budhaditya Deb, Chenguang Zhu, Ahmed Hassan Awadallah, and Dragomir Radev. Leveraging locality in abstractive text summarization. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6081–6093, 2022.

[131] Kyle Lo, Lucy Lu Wang, Mark Neumann, Rodney Kinney, and Daniel S Weld. S2orc: The semantic scholar open research corpus. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4969–4983, 2020.

[132] Agnes Lydia and Sagayaraj Francis. Adagrad—an optimizer for stochastic gradient descent. *International J. Inf. Comput. Sci*, 6(5):566–568, 2019.

[133] Veronica Lynn, Niranjan Balasubramanian, and H Andrew Schwartz. Hierarchical modeling for user personality prediction: The role of message-level attention. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5306–5316, 2020.

[134] Shutian Ma, Jin Xu, Jie Wang, and Chengzhi Zhang. NJUST @ CLSciSumm-17. In *Proceedings of the First Workshop on Scholarly Document Processing*, 2017.

[135] François Mairesse, Marilyn A Walker, Matthias R Mehl, and Roger K Moore. Using linguistic cues for the automatic recognition of personality in conversation and text. *Journal of Artificial Intelligence Research*, 30:457–500, 2007.

[136] Navonil Majumder, Soujanya Poria, Alexander Gelbukh, and Erik Cambria. Deep learning-based document modeling for personality detection from text. *IEEE Intelligent Systems*, 32(2):74–79, 2017.

[137] Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, 2014.

[138] Ziming Mao, Chen Henry Wu, Ansong Ni, Yusen Zhang, Rui Zhang, Tao Yu, Budhaditya Deb, Chenguang Zhu, Ahmed Awadallah, and Dragomir Radev. DYLE: Dynamic latent extraction for abstractive long-input summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1687–1698, 2022.

[139] Jose Maria Balmaceda, Silvia Schiaffino, and Daniela Godoy. How do personality traits affect communication among users in online social networks? *Online Information Review*, 38(1):136–153, 2014.

[140] Brian W Matthews. Comparison of the predicted and observed secondary structure of t4 phage lysozyme. *Biochimica et Biophysica Acta (BBA) - Protein Structure*, 405(2):442–451, 1975.

[141] Sandra C Matz, Michal Kosinski, Gideon Nave, and David J Stillwell. Psychological targeting as an effective approach to digital mass persuasion. *Proceedings of the National Academy of Sciences*, 114(48):12714–12719, 2017.

[142] Yash Mehta, Samin Fatehi, Amirmohammad Kazameini, Clemens Stachl, Erik Cambria, and Sauleh Eetemadi. Bottom-up and top-down: Predicting personality with psycholinguistic and language model features. In *2020 IEEE International Conference on Data Mining (ICDM)*, pages 1184–1189. IEEE, 2020.

[143] Yash Mehta, Navonil Majumder, Alexander Gelbukh, and Erik Cambria. Recent trends in deep learning based personality detection. *Artificial Intelligence Review*, pages 1–27, 2019.

[144] Robert Mercer. Locating and extracting key components of argumentation from scholarly scientific writing. *Dagstuhl Reports*, 6(4):3–15, 2016.

[145] Rada Mihalcea and Paul Tarau. Textrank: Bringing order into text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 404–411, 2004.

[146] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119, 2013.

[147] Santosh Kumar Mishra, Harshavardhan Kundarapu, Naveen Saini, Sriparna Saha, and Pushpak Bhattacharyya. IITP-AI-NLP-ML @ CL-SciSumm 2020, CL-LaySumm 2020, LongSumm 2020. In *Proceedings of the First Workshop on Scholarly Document Processing*, pages 270–276, 2020.

[148] Santosh Kumar Mishra, Naveen Saini, Sriparna Saha, and Pushpak Bhattacharyya. Scientific document summarization in multi-objective clustering framework. *Applied Intelligence*, 52(2):1520–1543, 2022.

[149] Makoto Miwa, Rune Sætre, Yusuke Miyao, and Jun'ichi Tsujii. A rich feature vector for protein-protein interaction extraction from multiple corpora. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 121–130, 2009.

[150] Yusuke Miyao, Kenji Sagae, Rune Sætre, Takuya Matsuzaki, and Jun'ichi Tsujii. Evaluating contributions of natural language parsers to protein–protein interaction extraction. *Bioinformatics*, 25(3):394–400, 2009.

[151] Fatemeh Mohades Deilami, Hossein Sadr, and Morteza Tarkhan. Contextualized multidimensional personality recognition using combination of deep neural network and ensemble learning. *Neural Processing Letters*, 54(5):3811–3828, 2022.

[152] Gurusamy Murugesan, Sabenabanu Abdulkadhar, and Jeyakumar Natarajan. Distributed smoothed tree kernel for protein-protein interaction extraction from the biomedical literature. *PLOS ONE*, 12:e0187379, 11 2017.

[153] I.B. Myers, L.K. Kirby, and K.D. Myers. *Introduction to Type: A Guide to Understanding Your Results on the Myers-Briggs Type Indicator*. Oxford Psychologists Press, 2000.

[154] Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In *Thirty-first AAAI Conference on Artificial Intelligence*, 2017.

[155] Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Çağlar Gùlçehre, and Bing Xiang. Abstractive text summarization using sequence-to-sequence RNNs and beyond. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290, 2016.

[156] Shashi Narayan, Shay B Cohen, and Mirella Lapata. Ranking sentences for extractive summarization with reinforcement learning. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1747–1759, 2018.

[157] Ernst H Nauta, Herman Mattie, and Wim RO Goslings. Effect of probenecid on the apparent volume of distribution and elimination of cloxacillin. *Antimicrobial Agents and Chemotherapy*, 6(3):300–303, 1974.

[158] Claire Nédellec. Learning language in logic-genic interaction extraction challenge. In *Proceedings of the Learning Language in Logic 2005 Workshop (LLL05)*, pages 31–37, 2005.

[159] Jeff Orlowski. The social dilemma. `https://www.thesocialdilemma.com/`, 2021.

[160] Daniel J. Ozer and Verónica Benet-Martínez. Personality and the prediction of consequential outcomes. *Annual Review of Psychology*, 57(1):401–421, 2006.

[161] Matteo Pagliardini, Prakhar Gupta, and Martin Jaggi. Unsupervised learning of sentence embeddings using compositional n-gram features. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 528–540, 2018.

[162] Raquel Mochales Palau and Marie-Francine Moens. Argumentation mining: The detection, classification and structure of arguments in text. In *Proceedings of the 12th International Conference on Artificial Intelligence and Law*, pages 98–107, 2009.

[163] Praveetha Patalay and Suzanne H Gage. Changes in millennial adolescent mental health and health-related behaviours over 10 years: a population cohort comparison study. *International Journal of Epidemiology*, 48(5):1650–1664, 2019.

[164] Yifan Peng and Zhiyong Lu. Deep learning for extracting protein-protein interactions from biomedical literature. In *BioNLP 2017*, pages 29–38, 2017.

[165] Yifan Peng, Chih-Hsuan Wei, and Zhiyong Lu. Improving chemical disease relation extraction with rich features and weakly labeled data. *Journal of Cheminformatics*, 8(1):1–12, 2016.

[166] James W Pennebaker and Laura A King. Linguistic styles: language use as an individual difference. *Journal of Personality and Social Psychology*, 77(6):1296, 1999.

[167] Jeffrey Pennington, Richard Socher, and Christopher D Manning. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

[168] Pierre Perruchet and Ronald Peereman. The exploitation of distributional information in syllable processing. *Journal of Neurolinguistics*, 17(2-3):97–119, 2004.

[169] Long N Phan, James T Anibal, Hieu Tran, Shaurya Chanana, Erol Bahadroglu, Alec Peltekian, and Grégoire Altan-Bonnet. SciFive: a text-to-text transformer model for biomedical literature. *arXiv preprint arXiv:2106.03598*, 2021.

[170] Jonathan Pilault, Raymond Li, Sandeep Subramanian, and Christopher Pal. On extractive and abstractive neural document summarization with transformer language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9308–9319, 2020.

[171] David J Pittenger. Cautionary comments regarding the Myers-Briggs type indicator. *Consulting Psychology Journal: Practice and Research*, 57(3):210, 2005.

[172] Sampo Pyysalo, Antti Airola, Juho Heimonen, Jari Björne, Filip Ginter, and Tapio Salakoski. Comparative analysis of five protein-protein interaction corpora. *BMC Bioinformatics*, 9(3):1–11, 2008.

[173] Sampo Pyysalo, Filip Ginter, Juho Heimonen, Jari Björne, Jorma Boberg, Jouni Järvinen, and Tapio Salakoski. BioInfer: a corpus for information extraction in the biomedical domain. *BMC Bioinformatics*, 8(1):1–24, 2007.

[174] Yifu Qiu and Shay B. Cohen. Abstractive summarization guided by latent hierarchical document structure. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5303–5317, 2022.

[175] Chanqin Quan, Lei Hua, Xiao Sun, and Wenjun Bai. Multichannel convolutional neural network for biological relation extraction. *BioMed Research International*, 2016, 2016.

[176] Dragomir R. Radev, Hongyan Jing, and Malgorzata Budzikowska. Centroid-based summarization of multiple documents: Sentence extraction, utility-based evaluation, and user studies. In *NAACL-ANLP 2000 Workshop: Automatic Summarization*, 2000.

[177] Dragomir R Radev, Pradeep Muthukrishnan, Vahed Qazvinian, and Amjad Abu-Jbara. The ACL anthology network corpus. *Language Resources and Evaluation*, 47(4):919–944, 2013.

[178] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020.

[179] Majid Ramezani, Mohammad-Reza Feizi-Derakhshi, and Mohammad-Ali Balafar. Knowledge graph-enabled text-based automatic personality prediction. *Computational Intelligence and Neuroscience*, 2022.

[180] Saichethan Reddy, Naveen Saini, Sriparna Saha, and Pushpak Bhattacharyya. IIITBH-IITP @ CL-SciSumm20, CL-LaySumm20, LongSumm20. In *Proceedings of the First Workshop on Scholarly Document Processing*, pages 242–250, 2020.

[181] Nils Reimers. Pretrained models. `https://www.sbert.net/docs/pretrained\textunderscoremodels.html`, 2020.

[182] Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, 2019.

[183] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "Why should I trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1135–1144, 2016.

[184] Anna Ritchie, Stephen Robertson, and Simone Teufel. Comparing citation contexts for information retrieval. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, pages 213–222, 2008.

[185] Brent W Roberts, Nathan R Kuncel, Rebecca Shiner, Avshalom Caspi, and Lewis R Goldberg. The power of personality: The comparative validity of personality traits, socioeconomic status, and cognitive ability for predicting important life outcomes. *Perspectives on Psychological Science*, 2(4):313–45, 2007.

[186] A David Rodrigues. *Drug-Drug Interactions*. CRC Press, 2nd edition, 2019.

[187] Stuart Rose, Dave Engel, Nick Cramer, and Wendy Cowley. Automatic keyword extraction from individual documents. *Text Mining: Applications and Theory*, 1:1–20, 2010.

[188] Sudipta Singha Roy, Sk Imran Hossain, MAH Akhand, and Kazuyuki Murase. A robust system for noisy image classification combining denoising autoencoder and convolutional neural network. *International Journal of Advanced Computer Science and Applications*, 9(1):224–235, 2018.

[189] Qian Ruan, Malte Ostendorff, and Georg Rehm. HiStruct+: Improving extractive text summarization with hierarchical structure information. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1292–1308, 2022.

[190] David L Sackett. Evidence-based medicine. *Seminars in Perinatology*, 21(1):3–5, 1997.

[191] Rune Sætre, Kenji Sagae, and Jun'ichi Tsujii. Syntactic features for protein-protein interaction extraction. In *2nd International Symposium on Languages in Biology and Medicine (Short Papers)*, volume 319 of *CEUR Workshop Proceedings*, 2007.

[192] Sunil Kumar Sahu and Ashish Anand. Drug-drug interaction extraction from biomedical texts using long short-term memory network. *Journal of Biomedical Informatics*, 86:15–24, 2018.

[193] Alexandra Samet. 2020 US SOCIAL MEDIA USAGE: How the coronavirus is changing consumer behavior. `https://www.businessinsider.com/2020-us-social-media-usage-report`, 2020.

[194] Robin M Schmidt. Recurrent neural networks (RNNs): A gentle introduction and overview. *arXiv preprint arXiv:1912.05911*, 2019.

[195] Hanna Schneider, Katrin Schauer, Clemens Stachl, and Andreas Butz. Your data, your vis: Personalizing personal data visualizations. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 10515, pages 374–392, 2017.

[196] Mike Schuster and Kuldip K Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681, 1997.

[197] Abigail See, Peter J Liu, and Christopher D Manning. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, 2017.

[198] Athar Sefid and C Lee Giles. SciBERTSUM: Extractive summarization for scientific documents. In *International Workshop on Document Analysis Systems*, pages 688–701. Springer, 2022.

[199] Isabel Segura-Bedmar, Paloma Martínez Fernández, and María Herrero Zazo. SemEval-2013 task 9: Extraction of drug-drug interactions from biomedical texts (DDIExtraction 2013). In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 341–350, 2013.

[200] Sudipta Singha Roy. *Investigating Citation Linkage as a Sentence Similarity Measurement Task using Deep Learning*. PhD thesis, Electronic Thesis and Dissertation Repository, The University of Western Ontario, 2020.

[201] Sudipta Singha Roy and Robert Mercer. Identifying protein-protein interaction using tree-transformers and heterogeneous graph neural network. In *The International FLAIRS Conference Proceedings*, volume 36, 2023.

[202] Sudipta Singha Roy and Robert E. Mercer. BioCite: A deep learning-based citation linkage framework for biomedical research articles. In *Proceedings of the 21st Workshop on Biomedical Language Processing*, pages 241–251, 2022.

[203] Sudipta Singha Roy and Robert E. Mercer. Building a synthetic biomedical research article citation linkage corpus. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 5665–5672, 2022.

[204] Sudipta Singha Roy and Robert E Mercer. Protein-Protein interaction extraction using attention-based tree-structured neural network models. In *The International FLAIRS Conference Proceedings*, volume 35, 2022.

[205] Sudipta Singha Roy and Robert E Mercer. Extracting drug-drug and protein-protein interactions from text using a continuous update of tree-transformers. In *The 22nd Workshop on Biomedical Natural Language Processing and BioNLP Shared Tasks*, pages 280–291, 2023.

[206] Sudipta Singha Roy and Robert E Mercer. Generating extractive and abstractive summaries in parallel from scientific articles incorporating citing statements. In *The 4th New Frontiers in Summarization Workshop (NewSumm)*, 2023.

[207] Sudipta Singha Roy, Robert E Mercer, and Souvik Kundu. Personality trait detection using an hierarchy of tree-transformers and graph attention network. In *Canadian Conference on Artificial Intelligence*, volume 36, 2023.

[208] Sudipta Singha Roy, Robert E Mercer, and Felipe Urra. Investigating citation linkage as a sentence similaritymeasurement task using deep learning. In *33th Canadian Conference on Artificial Intelligence*, 2020.

[209] S Sledzieski, R Singh, L Cowen, and B Berger. Sequence-based prediction of protein-protein interactions: a structure-aware interpretable deep learning model. In *Research in Computational Molecular Biology 25th Annual International Conference (RECOMB 2021)*, volume 25, 2021.

[210] Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. MPNet: Masked and permuted pre-training for language understanding. *Advances in Neural Information Processing Systems*, 33:16857–16867, 2020.

[211] Clemens Stachl, Ryan L Boyd, Kai T Horstmann, Poruz Khambatta, Sandra Matz, and Gabriella M Harari. Computational personality assessment-an overview and perspective. *PsyArXiv*, 2021.

[212] Ming-Hsiang Su, Chung-Hsien Wu, and Yu-Ting Zheng. Exploiting turn-taking temporal evolution for personality trait perception in dyadic conversations. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24(4):733–744, 2016.

[213] Xiangguo Sun, Bo Liu, Jiuxin Cao, Junzhou Luo, and Xiaojun Shen. Who am I? Personality detection based on deep learning for texts. In *2018 IEEE International Conference on Communications (ICC)*, pages 1–6, 2018.

[214] Kai Sheng Tai, Richard Socher, and Christopher D. Manning. Improved semantic representations from tree-structured long short-term memory networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1556–1566, 2015.

[215] Tommy Tandera, Derwin Suhartono, Rini Wongso, and Yen Lina Prasetio. Personality prediction system from facebook users. *Procedia Computer Science*, 116:604–611, 2017.

[216] Zhan Tang, Xuchao Guo, Zhao Bai, Lei Diao, Shuhan Lu, and Lin Li. A protein-protein interaction extraction approach based on large pre-trained language model and adversarial training. *KSII Transactions on Internet and Information Systems (TIIS)*, 16(3):771–791, 2022.

[217] Y. R. Tausczik and J. W. Pennebaker. The psychological meaning of words: LIWC and computerized text analysis methods. *Journal of Language and Social Psychology*, 29(1):24–54, 2010.

[218] Domonkos Tikk, Philippe Thomas, Peter Palaga, Jörg Hakenberg, and Ulf Leser. A comprehensive benchmark of kernel methods to extract protein–protein interactions from literature. *PLoS Computational Biology*, 6(7):e1000837, 2010.

[219] Marko Tkalčič, Berardina De Carolis, Marco de Gemmis, Ante Odic, and Andrej Košir. Preface: Empire 2014 – 2nd workshop on emotions and personality in personalized services. In *2nd Workshop on Emotions and Personality in Personalized Services (EMPIRE 2014)*, volume 1181 of *CEUR Workshop Proceedings*, pages 1–4. CEUR-WS.org, 2014.

[220] Anjana Umapathy, Karthik Radhakrishnan, Kinjal Jain, and Rahul Singh. CiteQA@CLSciSumm 2020. In *Proceedings of the First Workshop on Scholarly Document Processing*, pages 297–302, 2020.

[221] Niels Van de Ven, Aniek Bogaert, Alec Serlie, Mark J Brandt, and Jaap JA Denissen. Personality perception based on linkedin profiles. *Journal of Managerial Psychology*, 32(6):418–429, 2017.

[222] Sofie Van Landeghem, Yvan Saeys, Yves Van de Peer, and Bernard De Baets. Extracting protein-protein interactions from text using rich feature vectors and feature selection. In *3rd International Symposium on Semantic Mining in Biomedicine (SMBM 2008)*, pages 77–84. Turku Centre for Computer Sciences (TUCS), 2008.

[223] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 2017.

[224] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *Proceedings of the 6th International Conference on Learning Representations, ICLR*, 2018.

[225] Prashanth Vijayaraghavan, Eric Chu, and Deb Roy. DAPPER: Learning domain-adapted persona representation using pretrained BERT and external memory. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 643–652, 2020.

[226] Sarah Theres Völkel, Ramona Schödel, Daniel Buschek, Clemens Stachl, Quay Au, Bernd Bischl, Markus B"uhner, and Heinrich Hussmann. Opportunities and Challenges of Utilizing Personality Traits for Personalization in HCI: Towards a shared perspective from HCI and Psychology. In *Personalized Human-Computer Interaction*, chapter 2, pages 31–64. De Gruyter, 2019.

[227] Danqing Wang, Pengfei Liu, Yining Zheng, Xipeng Qiu, and Xuanjing Huang. Heterogeneous graph neural networks for extractive document summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6209–6219, 2020.

[228] Fei Wang, Kaiqiang Song, Hongming Zhang, Lifeng Jin, Sangwoo Cho, Wenlin Yao, Xiaoyang Wang, Muhao Chen, and Dong Yu. Salience allocation as guidance for abstractive summarization. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6094–6106, 2022.

[229] Yaushian Wang, Hung-Yi Lee, and Yun-Nung Chen. Tree transformer: Integrating tree structures into self-attention. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1061–1070, 2019.

[230] Yuker Wang, Victoria EH Carlton, George Karlin-Neumann, Ronald Sapolsky, Li Zhang, Martin Moorhead, Zhigang C Wang, Andrea L Richardson, Robert Warren, Axel Walther, et al. High quality copy number and genotype data from FFPE samples using molecular inversion probe (MIP) microarrays. *BMC Medical Genomics*, 2(1):8, 2009.

[231] Zhengjue Wang, Zhibin Duan, Hao Zhang, Chaojie Wang, Long Tian, Bo Chen, and Mingyuan Zhou. Friendly topic assistant for transformer based abstractive summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 485–497, 2020.

[232] Max Welling and Thomas N Kipf. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR 2017)*, 2016.

[233] WHO. COVID-19 disrupting mental health services in most countries, WHO survey. https://www.who.int/news/item/05-10-2020-covid-19-disrupting-mental-health-services-in-most-countries-who-survey, 2020.

[234] Wikipedia. Accuracy and precision. https://en.wikipedia.org/wiki/Accuracy_and_precision. Accessed: 2023-12-15.

[235] Wikipedia. Evaluation of binary classifiers. https://en.wikipedia.org/wiki/Evaluation_of_binary_classifiers. Accessed: 2023-12-15.

[236] Wikipedia. F-score. https:en.wikipedia.org/wiki/F-score. Accessed: 2023-12-15.

[237] Wikipedia. METEOR. https://en.wikipedia.org/wiki/METEOR. Accessed: 2023-12-15.

[238] Wikipedia. ROUGE. https://en.wikipedia.org/wiki/ROUGE_(metric). Accessed: 2023-12-15.

[239] Wikipedia. WordNet. https://en.wikipedia.org/wiki/WordNet. Accessed: 2023-12-15.

[240] Adina Williams, Nikita Nangia, and Samuel Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, 2018.

[241] Petra Wolffs, Halfdan Grage, Oskar Hagberg, and Peter Rådström. Impact of DNA polymerases and their buffer systems on quantitative real-time PCR. *Journal of Clinical Microbiology*, 42(1):408–411, 2004.

[242] Qianqian Xie, Jimin Huang, Tulika Saha, and Sophia Ananiadou. GRETEL: Graph contrastive topic enhanced language model for long document extractive summarization. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 6259–6269, 2022.

[243] Linzi Xing, Brad Hackinen, Giuseppe Carenini, and Francesco Trebbi. Improving context modeling in neural topic segmentation. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 626–636, 2020.

[244] Shweta Yadav, Asif Ekbal, Sriparna Saha, Ankit Kumar, and Pushpak Bhattacharyya. Feature assisted stacked attentive shortest dependency path based Bi-LSTM model for protein–protein interaction. *Knowledge-Based Systems*, 166:18–29, 2019.

[245] Shweta Yadav, Srivastsa Ramesh, Sriparna Saha, and Asif Ekbal. Relation extraction from biomedical and clinical text: Unified multitask learning framework. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2020.

[246] Liu Yang, Mingyang Zhang, Cheng Li, Michael Bendersky, and Marc Najork. Beyond 512 tokens: Siamese multi-depth transformer-based hierarchical encoder for long-form document matching. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 1725–1734, 2020.

[247] Tao Yang, Jinghao Deng, Xiaojun Quan, and Qifan Wang. Orders are unwanted: Dynamic deep graph convolutional network for personality detection. In *Proceedings of The Thirty-Seventh AAAI Conference on Artificial Intelligence (AAAI-23)*, pages 13896–13904, 2023.

[248] Tao Yang, Feifan Yang, Haolan Ouyang, and Xiaojun Quan. Psycholinguistic tripartite graph network for personality detection. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4229–4239, 2021.

[249] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489, 2016.

[250] Yu Yao, Xiuquan Du, Yanyu Diao, and Huaixu Zhu. An integration of deep learning with feature embedding for protein–protein interaction prediction. *PeerJ*, 7:e7126, 2019.

[251] Michihiro Yasunaga, Jungo Kasai, Rui Zhang, Alexander R Fabbri, Irene Li, Dan Friedman, and Dragomir R Radev. Scisummnet: A large annotated corpus and content-impact models for scientific paper summarization with citation networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7386–7393, 2019.

[252] Michihiro Yasunaga, Rui Zhang, Kshitijh Meelu, Ayush Pareek, Krishnan Srinivasan, and Dragomir Radev. Graph-based neural multi-document summarization. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 452–462, 2017.

[253] Jen-Yuan Yeh, Tien-Yu Hsu, Cheng-Jung Tsai, Pei-Cheng Cheng, and Jung-Yi Lin. On identifying cited texts for citances and classifying their discourse facets by classification techniques. *Journal of Information Science & Engineering*, 35(1), 2019.

[254] Han Yin, Yue Wang, Qian Li, Wei Xu, Ying Yu, and Tao Zhang. A network-enhanced prediction method for automobile purchase classification using deep learning. In *Twenty-Second Pacific Asia Conference on Information Systems (PACIS 2018)*, pages 111–126, 2018.

[255] Wu Youyou, Michal Kosinski, and David Stillwell. Computer-based personality judgments are more accurate than those made by humans. *Proceedings of the National Academy of Sciences*, 112(4):1036–1040, 2015.

[256] Kaixian Yu, Pei-Yau Lung, Tingting Zhao, Peixiang Zhao, Yan-Yuan Tseng, and Jinfeng Zhang. Automatic extraction of protein-protein interactions using grammatical relationship graph. *BMC Medical Informatics and Decision Making*, 18:35–43, 2018.

[257] Tiezheng Yu, Dan Su, Wenliang Dai, and Pascale Fung. Dimsum@ LaySumm 20: BART-based approach for scientific document summarization. In *Proceedings of the First Workshop on Scholarly Document Processing*, pages 303–309, 2020.

[258] Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. Big bird: Transformers for longer sequences. *Advances in Neural Information Processing Systems*, 33:17283–17297, 2020.

[259] Chrysoula Zerva, Minh-Quoc Nghiem, Nhung TH Nguyen, and Sophia Ananiadou. NaCTeM-UoM@ CL-SciSumm 2019. In *BIRNDL@ SIGIR*, pages 167–180, 2019.

[260] Hao Zhang, Renchu Guan, Fengfeng Zhou, Yanchun Liang, Zhi-Hui Zhan, Lan Huang, and Xiaoyue Feng. Deep residual convolutional neural network for protein-protein interaction extraction. *IEEE Access*, 7:89354–89365, 2019.

[261] Haopeng Zhang, Xiao Liu, and Jiawei Zhang. HEGEL: Hypergraph transformer for long document summarization. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022*, pages 10167–10176, 2022.

[262] Haopeng Zhang, Semih Yavuz, Wojciech Kryściński, Kazuma Hashimoto, and Yingbo Zhou. Improving the faithfulness of abstractive summarization via entity coverage control. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 528–535, 2022.

[263] Min Zhang, Jie Zhang, Jian Su, and Guodong Zhou. A composite kernel to extract relations between entities with both flat and structured features. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 825–832, 2006.

[264] Xingxing Zhang, Furu Wei, and Ming Zhou. HIBERT: Document level pre-training of hierarchical bidirectional transformers for document summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5059–5069, 2019.

[265] Yijia Zhang, Qingyu Chen, Zhihao Yang, Hongfei Lin, and Zhiyong Lu. BioWordVec, improving biomedical word embeddings with subword information and mesh. *Scientific Data*, 6(1):1–9, 2019.

[266] Yijia Zhang, Hongfei Lin, Zhihao Yang, and Yanpeng Li. Neighborhood hash graph kernel for protein–protein interaction extraction. *Journal of Biomedical Informatics*, 44(6):1086–1092, 2011.

[267] Yijia Zhang, Hongfei Lin, Zhihao Yang, Jian Wang, Shaowu Zhang, Yuanyuan Sun, and Liang Yang. A hybrid model based on neural networks for biomedical relation extraction. *Journal of Biomedical Informatics*, 81:83–92, 2018.

[268] Yufeng Zhang, Xueli Yu, Zeyu Cui, Shu Wu, Zhongzhen Wen, and Liang Wang. Every document owns its structure: Inductive text classification via graph neural networks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 334–339, 2020.

[269] Han Zhao, Zhengdong Lu, and Pascal Poupart. Self-adaptive hierarchical sentence model. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*, pages 4069–4076, 2015.

[270] Zhehuan Zhao, Zhihao Yang, Hongfei Lin, Jian Wang, and Song Gao. A protein-protein interaction extraction approach based on deep neural network. *International Journal of Data Mining and Bioinformatics*, 15(2):145–164, 2016.

[271] Ming Zhong, Pengfei Liu, Yiran Chen, Danqing Wang, Xipeng Qiu, and Xuan-Jing Huang. Extractive summarization as text matching. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6197–6208, 2020.

[272] Ming Zhong, Pengfei Liu, Danqing Wang, Xipeng Qiu, and Xuanjing Huang. Searching for effective neural extractive summarization: What works and what's next. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1049–1058, 2019.

[273] Qingyu Zhou, Nan Yang, Furu Wei, Shaohan Huang, Ming Zhou, and Tiejun Zhao. Neural document summarization by jointly learning to score and select sentences. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 654–663, 2018.

[274] Yangfu Zhu, Linmei Hu, Xinkai Ge, Wanrong Peng, and Bin Wu. Contrastive graph transformer network for personality detection. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence (IJCAI-22)*, pages 4559–4565, 2022.

[275] Anzhela Zhusupova. Characterizing the personality of Twitter users based on their timeline information. Master's thesis, Instituto Universitário de Lisboa, 2016.

Appendix A

Copyright Forms of the Papers

Copyright release forms for the published papers are included in this appendix.

Building a Synthetic Biomedical Research Article Citation Linkage Corpus

Sudipta Singha Roy, Robert E. Mercer

Abstract

Citations are frequently used in publications to support the presented results and to demonstrate the previous discoveries while also assisting the reader in following the chronological progression of information through publications. In scientific publications, a citation refers to the referenced document, but it makes no mention of the exact span of text that is being referred to. Connecting the citation to this span of text is called citation linkage. In this paper, to find these citation linkages in biomedical research publications using deep learning, we provide a synthetic silver standard corpus as well as the method to build this corpus. The motivation for building this corpus is to provide a training set for deep learning models that will locate the text spans in a reference article, given a citing statement, based on semantic similarity. This corpus is composed of sentence pairs, where one sentence in each pair is the citing statement and the other one is a candidate cited statement from the referenced paper. The corpus is annotated using an unsupervised sentence embedding method. The effectiveness of this silver standard corpus for training citation linkage models is validated against a human-annotated gold standard corpus.

Anthology ID: 2022.lrec-1.608
Volume: Proceedings of the Thirteenth Language Resources and Evaluation Conference
Month: June
Year: 2022
Address: Marseille, France
Editors: Nicoletta Calzolari, Frédéric Béchet, Philippe Blache, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Jan Odijk, Stelios Piperidis
Venue: LREC
SIG: –
Publisher: European Language Resources Association
Note: –
Pages: 5665–5672
Language: –
URL: https://aclanthology.org/2022.lrec-1.608
DOI: –
Bibkey: singha-roy-mercer-2022-building
Cite (ACL): Sudipta Singha Roy and Robert E. Mercer. 2022. Building a Synthetic Biomedical Research Article Citation Linkage Corpus. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 5665–5672, Marseille, France. European Language Resources Association.
Cite (Informal): Building a Synthetic Biomedical Research Article Citation Linkage Corpus (Singha Roy & Mercer, LREC 2022)
Copy Citation: BibTeX Markdown MODS XML Endnote More options…
PDF: https://aclanthology.org/2022.lrec-1.608.pdf

BioCite: A Deep Learning-based Citation Linkage Framework for Biomedical Research Articles

Sudipta Singha Roy, Robert E. Mercer

Abstract

Research papers reflect scientific advances. Citations are widely used in research publications to support the new findings and show their benefits, while also regulating the information flow to make the contents clearer for the audience. A citation in a research article refers to the information's source, but not the specific text span from that source article. In biomedical research articles, this task is challenging as the same chemical or biological component can be represented in multiple ways in different papers from various domains. This paper suggests a mechanism for linking citing sentences in a publication with cited sentences in referenced sources. The framework presented here pairs the citing sentence with all of the sentences in the reference text, and then tries to retrieve the semantically equivalent pairs. These semantically related sentences from the reference paper are chosen as the cited statements. This effort involves designing a citation linkage framework utilizing sequential and tree-structured siamese deep learning models. This paper also provides a method to create a synthetic corpus for such a task.

PDF Cite Search Video

Anthology ID: 2022.bionlp-1.23
Volume: Proceedings of the 21st Workshop on Biomedical Language Processing
Month: May
Year: 2022
Address: Dublin, Ireland
Editors: Dina Demner-Fushman, Kevin Bretonnel Cohen, Sophia Ananiadou, Junichi Tsujii
Venue: BioNLP
SIG: –
Publisher: Association for Computational Linguistics
Note: –
Pages: 241–251
Language: –
URL: https://aclanthology.org/2022.bionlp-1.23
DOI: 10.18653/v1/2022.bionlp-1.23
Bibkey: singha-roy-mercer-2022-biocite
Cite (ACL): Sudipta Singha Roy and Robert E. Mercer. 2022. BioCite: A Deep Learning-based Citation Linkage Framework for Biomedical Research Articles. In *Proceedings of the 21st Workshop on Biomedical Language Processing*, pages 241–251, Dublin, Ireland. Association for Computational Linguistics.
Cite (Informal): BioCite: A Deep Learning-based Citation Linkage Framework for Biomedical Research Articles (Singha Roy & Mercer, BioNLP 2022)
Copy Citation: BibTeX Markdown MODS XML Endnote More options…
PDF: https://aclanthology.org/2022.bionlp-1.23.pdf
Video: https://aclanthology.org/2022.bionlp-1.23.mp4

Protein-Protein Interaction Extraction using Attention-based Tree-Structured Neural Network Models

Sudipta Singha Roy
The University of Western Ontario

Robert E. Mercer
The University of Western Ontario

DOI: https://doi.org/10.32473/flairs.v35i.130660

ABSTRACT

In order to comprehend underlying biological processes, it is necessary to identify interactions between proteins. It is typically quite difficult to extract a protein-protein interaction (PPI) from text data as text data is complex in nature. Unlike sequential models, tree-structured neural network models have the ability to consider syntactic and semantic dependencies between different portions of the text and can provide structural information at the phrase level. This paper investigates tree-structured neural network models for the PPI task and the results show their supremacy over sequential models and their effectiveness for this task.

PDF

PUBLISHED

04-05-2022

HOW TO CITE

Singha Roy, S., & Mercer, R. E. (2022). Protein-Protein Interaction Extraction using Attention-based Tree-Structured Neural Network Models. *The International FLAIRS Conference Proceedings, 35*. https://doi.org/10.32473/flairs.v35i.130660

More Citation Formats

ISSUE

Vol. 35 (2022): Proceedings of FLAIRS-35

SECTION

Special Track: Neural Networks and Data Mining

LICENSE

Published by the LibraryPress@UF, an imprint of the University of Florida Press and the George A. Smathers Libraries at the University of Florida. | The Florida OJ service is provided through the FLVC Library Services and the George A. Smathers Libraries at the University of Florida. FLVC Privacy Policy.

ISSN 2334-0762

Identifying Protein-Protein Interaction using Tree-Transformers and Heterogeneous Graph Neural Network

Sudipta Singha Roy
The University of Western Ontario
https://orcid.org/0000-0003-2640-6300

Robert Mercer
The University of Western Ontario
https://orcid.org/0000-0002-0080-715X

DOI: https://doi.org/10.32473/flairs.36.133256

ABSTRACT

For a better understanding of the underlying biological mechanisms, it is crucial to identify the reciprocity between proteins. Often, extracting such interactions between proteins from biomedical articles faces challenges due to the complex sentence structure of the textual information sources. Most of the prominent previous works have applied additional hand-crafted features for the protein-protein interaction task. In this work, we have utilized two tree-structured attention-based neural network models along with a heterogeneous graph approach to perform this task. We suggest that the proposed model preserves the syntactic as well as the semantic information of the text. The experimental results demonstrate that even without using any additional feature extraction techniques, this model achieves significant performance boosts when applied on the five standard benchmark corpora compared to the previous works.

PDF

PUBLISHED

08-05-2023

HOW TO CITE

Singha Roy, S., & Mercer, R. . (2023). Identifying Protein-Protein Interaction using Tree-Transformers and Heterogeneous Graph Neural Network. *The International FLAIRS Conference Proceedings*, *36*(1). https://doi.org/10.32473/flairs.36.133256

ISSN 2334-0762

Extracting Drug-Drug and Protein-Protein Interactions from Text using a Continuous Update of Tree-Transformers

Sudipta Singha Roy, Robert E. Mercer

Abstract

Understanding biological mechanisms requires determining mutual protein-protein interactions (PPI). Obtaining drug-drug interactions (DDI) from scientific articles provides important information about drugs. Extracting such medical entity interactions from biomedical articles is challenging due to complex sentence structures. To address this issue, our proposed model utilizes tree-transformers to generate the sentence representation first, and then a sentence-to-word update step to fine-tune the word embeddings which are again used by the tree-transformers to generate enriched sentence representations. Using the tree-transformers helps the model preserve syntactical information and provide semantic information. The fine-tuning provided by the continuous update step adds improved semantics to the representation of each sentence. Our model outperforms other prominent models with a significant performance boost on the five standard PPI corpora and a performance boost on the one benchmark DDI corpus that are used in our experiments.

PDF | Cite | Search

Anthology ID: 2023.bionlp-1.25
Volume: The 22nd Workshop on Biomedical Natural Language Processing and BioNLP Shared Tasks
Month: July
Year: 2023
Address: Toronto, Canada
Editors: Dina Demner-fushman, Sophia Ananiadou, Kevin Cohen
Venue: BioNLP
SIG: –
Publisher: Association for Computational Linguistics
Note: –
Pages: 280–291
Language: –
URL: https://aclanthology.org/2023.bionlp-1.25
DOI: 10.18653/v1/2023.bionlp-1.25
Bibkey: singha-roy-mercer-2023-extracting
Cite (ACL): Sudipta Singha Roy and Robert E. Mercer. 2023. Extracting Drug-Drug and Protein-Protein Interactions from Text using a Continuous Update of Tree-Transformers. In *The 22nd Workshop on Biomedical Natural Language Processing and BioNLP Shared Tasks*, pages 280–291, Toronto, Canada. Association for Computational Linguistics.
Cite (Informal): Extracting Drug-Drug and Protein-Protein Interactions from Text using a Continuous Update of Tree-Transformers (Singha Roy & Mercer, BioNLP 2023)
Copy Citation: BibTeX | Markdown | MODS XML | Endnote | More options…
PDF: https://aclanthology.org/2023.bionlp-1.25.pdf

Interpretable Representation Learning for Personality Detection

Conference Proceedings: 2021 International Conference on Data Mining Workshops (ICDMW)
Author: Amirmohammad Kazemeini; Sudipta Singha Roy; Robert E. Mercer; Erik Cambria
Publisher: IEEE
Date: 7-10 Dec. 2021

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

BACK CLOSE WINDOW

The 36th Canadian Conference on Artificial Intelligence Canadian AI 2023
5—9 June 2023, Montréal

Personality Trait Detection using an Hierarchy of Tree-transformers and Graph Attention Network

Sudipta Singha Roy[†,*], Robert E. Mercer[†], Souvik Kundu[†]
[†] The University of Western Ontario

Abstract

Automatic personality trait detection from a person's writings is helpful for professionals to assess the mental health of an individual, as well as helping individuals to determine their strengths and weaknesses for making choices such as personal improvement, workplace compatibility, and life-style decision-making. Psychologists have identified a set of personality traits that may be present in an individual's personality. This work classifies the writings of an individual into a subset of these traits. The classifier model comprises an hierarchical structure of tree-transformers and a graph attention network (GAT). The tree-transformers encode the sentences and the following GAT layer encodes the complete text of an individual's writing. Our model has shown a large performance boost over two benchmark corpora compared to previous works.

Keywords: Personality traits, Dependency tree-transformer, Constituency tree-transformer, Graph attention network, Multi-label classification

1. Introduction

Artificial intelligence (AI) has become a valuable tool for aiding psychiatrists and healthcare professionals in addressing the growing incidence of mental health related issues and disorders [1]. This upward trajectory has garnered recent attention, with studies like "Changes in Mental Ill Health and Health-Related Behaviors in Two Cohorts of UK Adolescents" revealing that rates of depression symptoms as well as self-harm tendencies have risen to multiple times in 2015 compared to 2005 [2]. In addition, research has examined the effect of social media on mental health, including its impact on adolescents' mental health and the increasing prevalence of teen suicide [3].

The COVID-19 pandemic has exacerbated the rising incidence of mental health concerns. A Kaiser Family Foundation survey has reported that individuals have become more distressed and disconnected from their social life, with nearly 50% residents of America reporting that the pandemic has negatively impacted their mental wellbeing [4, 5].

A 2020 Harris Poll [6] shows social media usage has increased among US adults, about 50% reporting higher usage during the pandemic. This trend was particularly noticeable among younger age groups, with 60% of those aged 18 to 34, 64% of those aged 35 to 49, and 34% of those aged 65 and older reporting increased social media usage [7].

Personality traits refer to a collection of enduring qualities, rooted in psychological research [8], that define an individual's emotions and actions in a relatively consistent manner. The Big-Five personality traits (also called OCEAN) is the best accepted and most commonly used model of personality [9]. OCEAN describes personality with these five measures: Openness to Experience, Conscientiousness, Extraversion, Agreeableness, and Neuroticism (or positively keyed as emotional stability) [1]. Another frequently used personality model, the Myers-Briggs Type Indicator (MBTI) [10], categorizes 16 types of personalities characterized by a combination of four binary categories: Extroversion or Introversion, Sensing or

*singhar@uwo.ca

Intuition, Thinking or Feeling, and Judging or Perceiving. These traits play important roles in an individual's future and life outcomes [11, 12].

The upsurge in social media activity during the pandemic has resulted in more digital footprints being left behind. These footprints can reveal an individual's personality and emotional traits, as has been demonstrated by Kosinski et al. [13]. This presents an opportunity to leverage these data to provide tailored support to individuals based on their unique needs, thus transforming the pandemic challenge into a potential advantage in terms of mental health care.

Many countries have faced additional burden on their mental health services due to the COVID-19 pandemic, as highlighted by a survey conducted by the World Health Organization (WHO) [14]. Given the scarcity of mental health service resources and the surge in mental health issues, the rise in social media usage presents an window of opportunity for AI researchers to leverage the resulting digital footprints to aid in diagnosing individuals' mental health concerns.

Prior research has explored the connection between personality traits and mental health disorders. Several studies have evidenced that neuroticism is a crucial factor in the development of depression and anxiety disorders [15, 16]. In addition, studies have found that resilience is inversely correlated with neuroticism and positively associated with conscientiousness and extraversion. Moreover, The positive correlation between openness and resilience is modest, but significant statistically [17]. Thus, automatic comprehension of an individual's personality can have a significant impact on the treatment process for mental health concerns. This has the potential to improve treatment outcomes and alleviate the burden on mental health services.

In this study, we have developed two deep-learning models that integrate tree-transformers

Generating Extractive and Abstractive Summaries in Parallel from Scientific Articles Incorporating Citing Statements

Sudipta Singha Roy, Robert E. Mercer

Abstract

Summarization of scientific articles often overlooks insights from citing papers, focusing solely on the document's content. To incorporate citation contexts, we develop a model to summarize a scientific document using the information in the source and citing documents. It concurrently generates abstractive and extractive summaries, each enhancing the other. The extractive summarizer utilizes a blend of heterogeneous graph-based neural networks and graph attention networks, while the abstractive summarizer employs an autoregressive decoder. These modules exchange control signals through the loss function, ensuring the creation of high-quality summaries in both styles.

PDF
Cite
Search
Supplementary material
Supplementary material

Anthology ID: 2023.newsum-1.8
Volume: Proceedings of the 4th New Frontiers in Summarization Workshop
Month: December
Year: 2023
Address: Hybrid
Editors: Yue Dong, Wen Xiao, Lu Wang, Fei Liu, Giuseppe Carenini
Venue: NewSum
SIG: –
Publisher: Association for Computational Linguistics
Note: –
Pages: 75–86
Language: –
URL: https://aclanthology.org/2023.newsum-1.8
DOI: 10.18653/v1/2023.newsum-1.8
Bibkey: singha-roy-mercer-2023-generating
Cite (ACL): Sudipta Singha Roy and Robert E. Mercer. 2023. Generating Extractive and Abstractive Summaries in Parallel from Scientific Articles Incorporating Citing Statements. In *Proceedings of the 4th New Frontiers in Summarization Workshop*, pages 75–86, Hybrid. Association for Computational Linguistics.
Cite (Informal): Generating Extractive and Abstractive Summaries in Parallel from Scientific Articles Incorporating Citing Statements (Singha Roy & Mercer, NewSum 2023)
Copy Citation: BibTeX Markdown MODS XML Endnote More options…
PDF: https://aclanthology.org/2023.newsum-1.8.pdf
Supplementary ma… 2023.newsum-1.8.SupplementaryMaterial.zip
Supplementary ma… 2023.newsum-1.8.SupplementaryMaterial.txt

Curriculum Vitae

Name:	Sudipta Singha Roy
Post-Secondary Education and Degrees:	University of Western Ontario London ON, Canada Ph.D. Computer Science, 2020 - ongoing Supervisor: Dr. Robert E. Mercer University of Western Ontario London, ON, Canada Master's Studies in Computer Science, 2018-2020 Supervisor: Dr. Robert E. Mercer Khulna University of Engineering & Technology Khulna, Bangladesh B.Sc. Computer Science and Engineering, 2010-2014 Supervisor: Dr. M. M. A. Hashem
Honours and Awards:	Western Graduate Research Scholarship 2018-2023 Mitacs Accelerate Grant, 2020
Related Work Experience:	Graduate Research and Teaching Assistant The University of Western Ontario 2018-2023 Artificial Intelligence Intern Messagepoint, Toronto, ON, Canada September 2020 — December 2020. Lecturer, Department of Computer Science and Engineering Khulna University of Engineering & Technology, Khulna, Bangladesh November 2015 — August 2018.

Publications: Conference Papers

1. Sudipta Singha Roy and Robert E. Mercer, "Generating Extractive and Abstractive Summaries in Parallel from Scientific Articles Incorporating Citing Statements," *The 4th New Frontiers in Summarization Workshop (NewSumm)*, Singapore, 2023.

2. Sudipta Singha Roy and Robert E. Mercer, "Extracting Drug-Drug and Protein-Protein Interactions from Text using a Continuous Update of Tree-Transformers," *The 22nd Workshop on Biomedical Natural Language Processing and BioNLP Shared Tasks*, Toronto, Canada, 2023.

3. Sudipta Singha Roy and Robert E. Mercer, "Identifying Protein-Protein Interaction using Tree-Transformers and Heterogeneous Graph Neural Network," *The 36th International FLAIRS Conference Proceedings*, Florida, USA, 2023.

4. Sudipta Singha Roy, Robert E. Mercer and Souvik Kundu, "Personality Trait Detection using an Hierarchy of Tree-transformers and Graph Attention Network," *The 36th Canadian Conference on Artificial Intelligence*, Montreal, Canada, 2023.

5. Sudipta Singha Roy and Robert E. Mercer, "Protein-Protein Interaction Extraction using Attention-based Tree-Structured Neural Network Models," *The 35th International FLAIRS Conference Proceedings*, Florida, USA, 2022.

6. Sudipta Singha Roy and Robert E. Mercer, "BioCite: A Deep Learning-based Citation Linkage Framework for Biomedical Research Articles," *Proceedings of the 21st Workshop on Biomedical Language Processing*, Dublin, Ireland, 2022.

7. Sudipta Singha Roy and Robert E. Mercer, "Building a Synthetic Biomedical Research Article Citation Linkage Corpus," *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, Marseille, France, 2022.

8. Nicholas Elder, Robert E. Mercer and Sudipta Singha Roy, "Building a Synthetic Biomedical Research Article Citation Linkage Corpus," *Proceedings of The 16th Lingusitic Annotation Workshop (LAW-XVI) within LREC2022*, Marseille, France, 2022.

9. Amirmohammad Kazemeini, Sudipta Singha Roy, Robert E. Mercer and Erik Cambria, "Building a Synthetic Biomedical Research Article Citation Linkage Corpus," *2021 International Conference on Data Mining Workshops (ICDMW)*, Auckland, New Zealand, 2021.

10. Sudipta Singha Roy, Robert E. Mercer and Felipe Urra, "Investigating Citation Linkage as a Sentence Similarity Measurement Task Using Deep Learning," *The 33th Canadian Conference on Artificial Intelligence*, Ottawa, Canada, 2020.

Under Review: Conference Papers

1. Sudipta Singha Roy and Robert E. Mercer, "Investigating Semantic Similarity-Induced Parallel Training of Abstractive and Extractive Scientific Document Summarizers"

2. Sudipta Singha Roy and Robert E. Mercer, "Enhancing Scientific Document Summarization with Research Community Perspective and Background Knowledge"

3. Sudipta Singha Roy and Robert E. Mercer, "Detecting Personality Traits from Texts using an Hierarchy of Tree-Transformers and Graph Attention Network with Word Embedding Refinement"