


People's Democratic Republic of Algeria

Ministry of Higher Education and Scientific Research

AKLI MOHAND OULHADJ UNIVERSITY -BOUIRA-

Faculty of Sciences and Applied Sciences

Computer Science Department

Master Thesis

In Computer Science

Speciality : *Information Systems and Software Engineering -ISIL-*

Theme

Relation Extraction from Biomedical Texts using Deep Learning

---

Supervised by

— Dr. AID Aicha

Realized by

— ABDELLAOUI Imane

— AZZOUNE Rima

2023/2024

Acknowledgments

First and foremost, praises, and thanks to the God for giving us patience, health and courage to carry out this work.

We extend our deep gratitude and sincere thanks to our supervisor, ***Dr. Aid.A****, for her efforts and important and effective role in making this research a success. We are very grateful for her constant encouragement and belief in us and our abilities, and for her presence, advice, and loyalty to us throughout this research.*

Our special thanks go to the entire faculty and staff of ***the Computer Science Department*** *at* ***AKLI MOHAND OULHADJ UNIVERSITY -BOUIRA-****. Their support, both academic and administrative, has provided a conducive environment for our research.*

We are also deeply indebted to our colleagues and friends, who have always been there to lend a listening ear and provide constructive feedback. Their camaraderie and support have made this journey a rewarding experience.

Last but not least, our deepest gratitude goes to our families. Their unwavering love, patience, and understanding have been the backbone of our efforts. Their sacrifices and encouragement have driven us to overcome challenges and strive for excellence.

-Abdellaoui Imane.

-Azzoune Rima.

Dedications

This modest work is the fruit of five hard but wonderful years of study.

I dedicate it first and foremost to the people without whom all this would not have been possible. The people who made me who I am today, those who shared my joys, my laughter, my tears, and who had to endure my moods. People who lived the longest after these five years (and much more), those which, without their limitless encouragement, their trust, their love, their presence and their advice, everything that I would have accomplished in my life would have no meaning. To my role models in life, Mom and Dad, it is in your pride that I draw my strength.

My sisters, my brothers who have always encouraged me and never stopped believing in me. To all my family who has always been there for me. You are my source of motivation.

To Rima, my lifelong partner, whose determination and commitment have been a constant source of inspiration. Thank you for your valuable collaboration and your unfailing support throughout this journey. We overcame challenges and reached new heights. To our friendship and future success, engraved in these lines and beyond.

To my friends, my relatives and all the people who participated directly or indirectly in the realization of this project. I can't thank you enough for everything you do for me.

Abdellaoui Imane.

Dedications

I dedicate this modest work to

My dear parents, who have always done their best and supported me throughout my life. For the sacrifices you have made, your patience, love, and unwavering trust in me, I am eternally grateful.

My beloved brothers and sisters, for their constant love and unlimited support.

My dear partner Imane, for her hard work and unwavering support. I wish her all the best.

Myself, for showing up, pushing through, and finishing what I started.

All my teachers since my first years of studies, and to all those who feel dear to me and whom I have failed to mention.

Azzoune Rima.

Abstract

Motivation

Relation extraction (RE) in the biomedical domain is crucial due to the enormous volume of textual data generated by scientific research and clinical practices. Biomedical publications, patient reports, and clinical databases contain valuable information that, when properly structured and interpreted, can revolutionize diagnostics, treatments, and disease understanding. Manually extracting these relations is time-consuming and prone to errors. Traditional relation extraction techniques have limitations in terms of precision and scalability. Recent advances in deep learning models, particularly Transformers, offer new perspectives by enabling the analysis and extraction of information with unprecedented precision and speed. These advances make biomedical information more accessible and useful for healthcare professionals and researchers, thus improving clinical decision support tools and reducing the time required to find relevant information.

Objectives

This work focuses on the application of Transformers techniques, specifically SciBERT models, to extract relation from biomedical texts. The main objective is to develop a system capable of identifying and structuring relations between biomedical entities with high precision.

Results

In our results, the proposed model, RE-SciBERT, demonstrated exceptional performance thanks to a rigorous fine-tuning process. By applying advanced fine-tuning techniques to the SciBERT model, we optimized the hyperparameters and improved the precision of biomedical relation extraction. The results show that our model achieved a remarkable F1 score of 90.10, with a precision of 77.15 and a recall of 75.73, thus surpassing the performance of the compared models.

Keywords

Relation Extraction, Biomedical Domain, Clinical Texts, Transformers, SciBERT.

Résumé

Motivation

L'extraction des relations (RE) dans le domaine biomédical est cruciale en raison de l'énorme volume de données textuelles issues de la recherche scientifique et des pratiques cliniques. Les publications biomédicales, les rapports de patients et les bases de données cliniques contiennent des informations précieuses qui, correctement structurées et interprétées, peuvent révolutionner les diagnostics, les traitements et la compréhension des maladies. L'extraction manuelle de ces relations est chronophage et sujette à des erreurs. Les techniques traditionnelles d'extraction de relations ont des limites de précision et de scalabilité. Les récents progrès des modèles d'apprentissage profond, notamment les Transformers, offrent de nouvelles perspectives en permettant une analyse et une extraction d'informations avec une précision et une rapidité sans précédent. Ces avancées rendent les informations biomédicales plus accessibles et utiles pour les professionnels de santé et les chercheurs, améliorant ainsi les outils de support à la décision clinique et réduisant le temps nécessaire pour trouver des informations pertinentes

Objectifs

Ce travail se concentre sur l'application des techniques Transformers, spécifiquement les modèles SciBERT, pour extraire des relations à partir de textes biomédicaux. L'objectif principal est de développer un système capable d'identifier et de structurer les relations entre les entités biomédicales avec une grande précision.

Résultats

Dans nos résultats, le modèle proposé, RE-SciBERT, a démontré une performance exceptionnelle grâce à un processus de fine-tuning rigoureux. En appliquant des techniques avancées de fine-tuning sur le modèle SciBERT, nous avons optimisé les hyperparamètres et amélioré la précision de l'extraction des relations biomédicales. Les résultats montrent que notre modèle atteint un score F1 remarquable de 90.10, avec une précision de 77,15 et un rappel de 75,73, surpassant ainsi les performances des modèles comparés.

Mots-clés

Extraction des relations, Domaine biomédical, Textes cliniques, Transformers, SciBERT.

Contents

List of Figures

List of Tables

List of Acronyms

RE	Relation Extraction.
NLP	Natural Language Processing.
DocRE	Document-level RE.
DL	Deep Learning.
BioRE	Biomedical Relation Extraction.

General introduction

The continuous growth of biomedical literature poses unprecedented challenges for information retrieval and data mining techniques. Repositories like pubMed central and MEDLINE are experiencing an exponential surge in publications, demanding innovative solutions to transform this data into actionable knowledge. Natural language processing emerges as a powerful tool in this endeavor, empowering researchers to interpret and process text for human-understandable knowledge extraction. Relation extraction systems, a specific type of NLP tool, offer a solution to this critical need.

Biomedical Relation Extraction focuses on automatically identifying relationships between chemical compounds, genes, and proteins. Unveiling these relationships is crucial for knowledge extraction, ultimately paving the way for novel treatment development and disease cause identification. While BioRE offers immense potential, it faces significant challenges due to the inherent complexities of biomedical texts. These challenges include specialized vocabulary, diverse terminologies, and intricate sentence structures.

One promising solution lies in fine-tuning pre-trained transformer-based RE systems. Transformers, a type of neural network architecture, have demonstrated remarkable effectiveness across various NLP tasks, including RE. Their ease of use and training efficiency make them a preferred choice to other models. Additionally, transformers offer a fertile ground for continuous research and advancements, ensuring ongoing performance improvements.

This study delves into the application of transformer models specifically for BioRE tasks. We leverage NLP and Deep Learning techniques to train a SciBERT model for chemical-protein relationship extraction within the biomedical domain. Our proposed model builds upon the pre-trained language comprehension capabilities of SciBERT. Fine-

tuning this model on the ChemProt dataset equips it with the necessary knowledge of general RE patterns, ultimately enhancing its capacity for accurate chemical-protein relation extraction in biomedical texts.

This dissertation is structured into four distinct chapters :

- In the first chapter, we establish a comprehensive foundation in RE. We'll explore the most important approaches, techniques, and datasets used to model and benchmark RE systems. This chapter will also delve into the evaluation metrics that are crucial for assessing their performance.
- In the second chapter, various preliminary concepts are introduced, followed by an in-depth exploration of advancements in artificial intelligence. The focus is particularly on DL and neural networks. The chapter further examines the application of neural networks in NLP, with a special emphasis on transformers. Additionally, it includes a comprehensive analysis of related works.
- The third chapter outlines the proposed approach to address the problem. It presents the datasets utilized, the models developed, and the architecture of the system.
- The last chapter wraps things up by analyzing and discussing the research findings we presented throughout this work. We'll delve into the specifics of our experimental setup, including the software used. Additionally, we'll dissect the results, showcasing the performance gains achieved through our model training process.

Chapter 1

Relation Extraction

1.1 Introduction

Relation extraction serves as a crucial component in natural language processing and data analysis, enabling the identification and characterization of associations that exist between diverse entities. This process involves parsing through textual or structured data to unveil the underlying relationships that may be implicit or explicit. The objective is to distill meaningful patterns and connections, contributing to a more profound comprehension of the information at hand.

In the subsequent sections, we will provide a comprehensive of relation extraction, this will include discussions on its definition. Subsequently, we will delve into specific aspects of relation extraction, including types, extraction pipeline, applications, approaches, biomedical aspects, datasets, and evaluation metrics.

1.2 Relation Extraction

Relation extraction (RE) is a fundamental task in Natural Language Processing (NLP) that aims to automatically identify and classify semantic relationships between entities mentioned in text [8]. In the context of relation extraction, entities are typically real-world objects such as people, organizations, locations, or other named entities.

Example : "Apple Inc. announced the acquisition of a startup company". - Here, the extracted relation is "organization-action-object", where "Apple Inc". (organization) is the entity performing the action "announce", and the object of this action is "the

acquisition of a startup company" (see Figure 1.1).

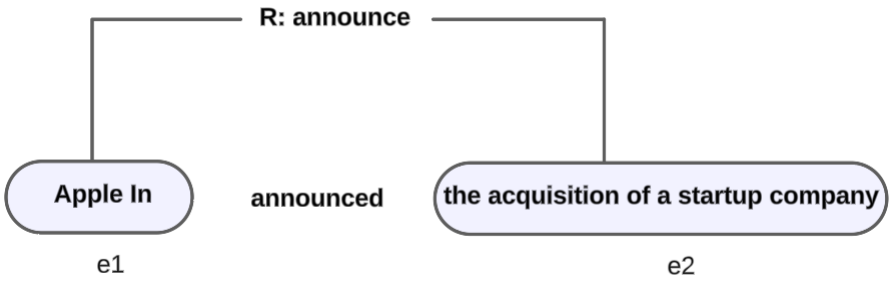


FIGURE 1.1 – Example of relation extraction.

1.2.1 Relation Types

In the task of Relation Extraction, relation types refer to the different types of semantic relationships that exist between entities mentioned in text. These relations can vary depending on the number of entities they link, the granularity and scope of the relations extracted, the domain, or the specific application. Some common relation types in Relation Extraction include [9] :

- **Binary relations :**, A binary relation is a relation involving two entities (which can be called also arguments of the relation) (e.g. : "The company Apple was founded by Steve Jobs." e1 :Apple "Organization", e2 :Steve Jobs "Person", Relation : Founded By).
- **N-ary relations :** They link three or more entities. They are good for verbs which can take multiple arguments or for event representation [9], (e.g. : "The collaboration between Microsoft, Intel, and Dell led to the development of a groundbreaking technology." e1 : Microsoft "Organization", e2 : Intel "Organization", e3 : Dell "Organization", Relation : Collaboration.)
- **First-order relations :** They connect two or more entities.
- **Higher-order relations** They link an entity with one or many relations.
- **Targeted – Predefined relations :** They are part of a set of predefined known relations ; relations that are already established or specified in advance. Thus, the task of their extraction bears resemblance to a classification task.

- **Emergent relations :** They are relations that are not predefined beforehand and may be discovered or identified during the process. The approach diverges from assuming a predefined set of relations. Instead, it aims to extract all possible relations in an unsupervised or semi-supervised manner. This concept aligns with Open Information Extraction (OpenRE), which represents an extraction paradigm designed to address an unlimited range of relations.
- **Mention-level** also known as **sentence-level RE**. The goal of sentence-level RE is to identify the relationship between two entities that are mentioned together in the same sentence [8]. The relationships are identified and classified based on the context and semantic meaning of the sentence in which the entities appear.
- **Global level**, also known as **document-level RE**, aims to determine the relation between two entities from a document of multiple sentences [10]. The goal of document-level relation extraction is to understand the connections and interactions between these entities within the context of the entire document. This task is more complex than sentence relation extraction, as it requires an understanding of the broader context and relationships between entities that may not be explicitly stated in a single sentence.

In this work, we focus on document-level Relation Extraction. This type of RE focuses on extracting relations from documents that are relevant to a specific domain or industry. For example, in the field of biomedical research, one might be interested in extracting relations between genes, proteins, and diseases. In this case, the extracted relations would be specific to the biomedical domain.

1.2.2 Relation Extraction Pipeline

Relation extraction entails multiple phases within a pipeline approach. This methodology involves the systematic breakdown of the relation extraction task into various sub-tasks. These sub-tasks encompass different aspects of the extraction process, and can be described as follows :

Entity recognition

Entity recognition, also known as Named Entity Recognition (NER), is a fundamental task in natural language processing that encompasses identifying and classifying entities

in a text. The original design idea for NER was to parse the text, to identify proper nouns from the text, and to categorize them.

Relation identification

Relation identification involves determining the specific relationship between entities mentioned in a sentence or text. By using natural language processing techniques, such as dependency parsing and pattern matching, relation identification aims to understand the syntactic and semantic structure of the sentence. It helps us recognize and classify the relationship between entities, such as "is associated with," "causes", "treats", or "interacts with".This process is essential for extracting structured information from unstructured text, enabling us to uncover hidden connections and gain deeper insights.

1.2.3 Relation Extraction Applications

Relation Extraction has transformed from a technical feat to a transformative tool, unlocking valuable knowledge hidden within textual data. Its applications span an impressive range of fields, each leveraging its ability to identify and classify relationships between entities mentioned in text. Let's take an in-depth look at some prominent examples from different domains :

Knowledge Graph Construction

Relation extraction is an essential component in constructing knowledge graphs. One notable system in this domain is Open IE (Open Information Extraction), proposed by Banco et al.[11]. Open IE aims to extract relations in an unsupervised manner from large amounts of text. Another notable work is Reverb, introduced by Fader et al.[12], which focuses on extracting binary relations from text using a pattern-based approach. These systems have been instrumental in populating knowledge graphs with structured information, facilitating advanced knowledge representation and semantic search.

Biomedical Research

Relation extraction is of great importance in biomedical research, particularly in extracting relationships between biomedical entities. One notable system is BioBERT (Biomedical Bidirectional Encoder Representations from Transformers), proposed by Lee et

al.[13]. BioBERT is a domain-specific language model fine-tuned on a large corpus of biomedical literature, enabling it to capture complex relationships between genes, proteins, diseases, and drugs. Another system is DNorm, introduced by Leaman et al.[14], which focuses on normalizing disease mentions in biomedical texts by identifying relationships between disease names and their corresponding unique identifiers.

Opinion Mining and Sentiment Analysis

Relation extraction is employed in opinion mining and sentiment analysis to identify relationships between entities and sentiments expressed in text. One notable system is the SenticNet framework proposed by Cambria et al.[15]. SenticNet combines natural language processing and machine learning techniques to extract and analyze relationships between entities and their associated sentiments, enabling fine-grained sentiment analysis in various domains.

Social Network Analysis

Relationship extraction is used in social network analysis to uncover connections and relationships between individuals or entities within a social network. One notable system is Stanford's CoreNLP, which provides a suite of natural language processing tools, including relation extraction capabilities. CoreNLP has been widely used for extracting relationships such as family relations, friendships, and professional connections from social media data[16].

Financial Analysis

Relation extraction is applied in financial analysis to extract relationships between companies, individuals, and financial events mentioned in textual sources. One notable system is FinBERT, introduced by Yang et al.[17], which is a domain-specific language model fine-tuned on financial texts, enabling it to capture relationships and sentiments related to financial entities.

Question Answering Systems

Relation extraction is employed in question answering systems to understand and answer questions that require knowledge about relationships between entities. One notable

system is OpenQA (Open Question Answering), proposed by Yates et al.[18]. OpenQA uses relation extraction techniques to identify relevant relationships between entities in text, enabling it to provide accurate and informative answers to user queries.

1.3 Relation Extraction Approaches

Relation extraction has witnessed a transformative evolution in methodologies, driven by the increasing complexity of biomedical texts. This section provides an overview of the diverse approaches employed for extracting relationships between entities within literature, with a focus on the progression from traditional methods to advanced deep learning techniques.

Traditional Approaches

Traditional approaches to relation extraction relied on handcrafted features and rule-based systems. While these methods have been effective in some domains, they often require manual feature engineering and domain-specific knowledge. In our work, traditional approaches may not be suitable due to the need for scalability in DocRE.

Therefore, we will focus in our proposed solution on exploring more advanced techniques that can overcome the limitations of traditional approaches and efficiently extract relationships from biomedical texts.

Semi-Supervised and Unsupervised Learning

Semi-supervised and unsupervised approaches to relation extraction aim to alleviate the reliance on labeled data by leveraging unlabeled or partially labeled text corpora. To train models, semi-supervised methods typically include a small portion of labeled data and a larger pool of unlabeled data [19].

Unsupervised methods use techniques like clustering, co-occurrence analysis, or graph-based algorithms to extract relationships from unlabeled text. These approaches have the potential to be scalable and flexible, but they may not perform as well as supervised methods in complex or nuanced relation extraction tasks.

Deep Learning Approaches

Deep learning has revolutionized relation extraction by enabling models to automatically learn hierarchical representations of text data [20]. Deep learning architectures, such as Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), or transformer-based models like BERT (Bidirectional Encoder Representations from Transformers), have demonstrated remarkable performance in relation extraction tasks. These models can capture complex linguistic patterns and contextual information, leading to state-of-the-art results in various relation extraction benchmarks. The proposed work aims to leverage the capabilities of deep learning models to improve relation extraction in the biomedical context.

Hybrid Approaches

The combined strength of multiple techniques, such as rule-based, supervised, and deep learning methods, is leveraged in hybrid approaches [21]. In a hybrid system, a supervised learning model and a deep learning model may be employed to classify the candidates, while using rule-based heuristics to generate candidate relations. Hybrid approaches incorporate diverse methodologies to offer flexibility and robustness, but may need more computational resources and expertise for development and deployment.

1.4 Biomedical Relation Extraction

Biomedical Relation Extraction stands as a revolutionary force in deciphering the complexities of life sciences research. It delves into the vast realm of biomedical literature, not just reading the words, but actively unveiling the relationships that bind entities like genes, proteins, diseases, and drugs. These relationships, intricate and diverse, can shed light on interactions, regulations, pathways, and the very fabric of biological processes.

Biomedical relation extraction involves automatically identifying and extracting relationships between biomedical entities mentioned in textual sources, such as scientific articles, clinical records, and biomedical databases.

In biomedical texts, Relation Extraction models aim to identify and extract various types of relationships between entities. The relationships can span across different do-

mains, including drug interactions, protein-protein interactions, gene-disease associations, and more. Here are some common types of relationships that can be extracted from biomedical texts :

Chemical-Protein Relations

Chemical-protein interactions are vital for understanding how small molecules, such as drugs, interact with proteins to influence biological processes. These interactions are crucial in drug design, toxicology, and elucidating disease mechanisms. By extracting these relationships from biomedical texts, researchers can identify potential drug targets, comprehend adverse drug reactions, and understand mechanisms of action. Examples include drug binding (e.g., statins to HMG-CoA reductase), enzyme inhibition (e.g., organophosphates inhibiting acetylcholinesterase), and protein modulation via allosteric sites [7].

Gene-Disease Associations

Identifying relationships between genes and diseases is crucial for understanding the genetic basis of diseases and identifying potential therapeutic targets. For example, identifying associations between the BRCA1 gene and breast cancer sheds light on the genetic predisposition to the disease [22].

Protein-Protein Interactions (PPIs)

Proteins often interact with each other to perform various biological functions. Extracting protein-protein interactions from literature aids in deciphering complex cellular processes and signaling pathways. For instance, identifying interactions between p53 and MDM2 proteins elucidates their role in regulating cell growth and apoptosis [23].

Drug-Target Interactions

Understanding the interactions between drugs and their molecular targets is essential for drug discovery and development. Extracting drug-target interactions from literature assists in identifying potential drug candidates and predicting drug efficacy and side effects. For example, identifying the interaction between aspirin and cyclooxygenase-1 (COX-1) informs its mechanism of action in inhibiting platelet aggregation[24].

Drug-Drug Interactions

Drug-drug Interactions (DDIs) represent significant challenges in pharmacotherapy, with the potential to lead to adverse outcomes or therapeutic failures. These interactions occur when two or more drugs interact, altering their pharmacokinetic or pharmacodynamic properties. A notable example is the interaction between warfarin and ciprofloxacin, where ciprofloxacin inhibits warfarin metabolism, leading to increased anticoagulant effects and bleeding risk [25].

Virus-Host Interactions

Virus-host interactions refer to the intricate relationships between viruses and their host organisms. Understanding these interactions is crucial for comprehending viral pathogenesis, developing antiviral strategies, and advancing our knowledge of host immune responses. For example, interaction between severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2) and human host cells in COVID-19 [26].

Biomedical relation extraction finds diverse applications across various domains in biomedical research and healthcare. Several systems and approaches have been developed to address specific applications, including :

BioBERT and SciBERT

These are specialized language models pre-trained on large-scale biomedical text data. They capture the domain-specific knowledge needed for biomedical relation extraction tasks and have shown improved performance in various relation extraction challenges, [13][27].

BioNLP Shared Task

The BioNLP shared task series organizes challenges to promote the development of relation extraction methods for specific biomedical relationships. Examples include protein-protein interactions, gene-disease associations, drug-drug interactions, and more. Participating in these challenges helps advance the field[28][29].

Pharmacovigilance and Adverse Event Reporting Systems

Pharmacovigilance systems collect and analyze data on Adverse drug events (ADEs) reported by healthcare professionals and patients. These systems help identify previously unknown drug-drug interactions and assess their impact on patient safety. By monitoring ADE reports, regulatory authorities and pharmaceutical companies can take necessary actions to mitigate risks associated with drug-drug interactions [30].

Clinical Decision Support Systems

Biomedical relation extraction plays a crucial role in clinical decision support systems by extracting relationships between patient characteristics, diagnoses, treatments, and outcomes. This aids in personalized medicine, treatment recommendations, and clinical research [31].

Pathway Reconstruction Systems

Pathway reconstruction systems aim to extract relationships between genes, proteins, and biological processes to reconstruct intricate biological pathways. Examples of such systems include PathwayStudio [32] and ingenuity pathway analysis [33], which leverage biomedical relation extraction to map interactions between molecular entities and elucidate signaling cascades and metabolic pathways.

Integrating extracted relationships into Decision Support Systems (DSS) in a clinical setting can greatly enhance the capabilities of healthcare professionals and improve patient outcomes. Here are some details and practical examples to delve into :

Practical Examples

- **Alerting for Potential Drug Interactions :** When a physician enters a patient's medication list into a DSS, the system can automatically flag potential drug interactions, providing recommendations to avoid adverse effects.
- **Symptom-Based Diagnosis Assistance :** DSS can analyze a patient's reported symptoms, compare them with known disease-symptom associations, and generate a ranked list of potential diagnoses, aiding the clinician in decision-making.

- **Treatment Recommendations :** Based on a patient's medical condition and characteristics, a DSS can suggest the most effective treatment options by considering drug-condition associations and treatment-efficacy relationships.
- **Personalized Risk Assessment :** By incorporating disease-risk factor associations, DSS can assess a patient's risk of developing certain conditions and recommend appropriate preventive measures, such as lifestyle modifications or screenings.
- **Predictive Analytics for Disease Progression :** By utilizing disease-risk factor associations and patient data, DSS can predict the likelihood of disease progression or deterioration. This information can assist healthcare professionals in developing proactive treatment plans and interventions to slow or manage disease progression effectively.
- **Treatment Adherence Monitoring :** DSS can extract and analyze relationships between medication adherence and patient outcomes. By integrating this information, healthcare professionals can monitor patients' adherence to prescribed treatments and intervene if non-compliance is detected, improving treatment effectiveness.

These examples illustrate how integrating extracted relationships into DSS can support healthcare professionals in making informed decisions, improving patient care, and reducing the likelihood of errors or adverse events.

1.5 Biomedical RE Datasets

A benchmark dataset commonly used in the research community can be used to evaluate the performance of a Relation Extraction model. In the following section, we present some important and reliable datasets that are commonly used to report outcomes related to RE modeling.

BioCreative VI ChemProt

The BioCreative VI ChemProt corpus is a valuable resource for studying chemical-protein interactions in scientific literature. It contains 6,500 interactions across 1,820 PubMed abstracts, providing standardized annotations, diverse interaction types, and

additional contextual information. It has potential applications in drug discovery, protein function understanding, and personalized medicine [7].

Biomedical Relation Extraction Dataset (BioRED)

BioRED is a dataset created for relation extraction in the biomedical field. It focuses on identifying connections between various biomedical entities such as genes, proteins, diseases, and chemicals. The dataset consists of annotated sentences or abstracts from the biomedical literature, with labels indicating the relationships between entities. In total, there are 20,419 entities mentioned in the BioRED corpus, representing 3,869 unique concept identifiers [34].

BioInfer

BioInfer is a project and dataset focused on biomedical text mining and Natural Language Processing. It provides a collection of scientific abstracts and annotations for extracting biomedical information. The dataset has been widely used to develop and evaluate techniques for information extraction, entity recognition, and relationship extraction in the biomedical domain. BioInfer has played a significant role in advancing the field of biomedical text mining [35].

BioCreative V CDR

The BioCreative V Chemical-Disease Relation (BC5CDR) corpus, a cornerstone in biomedical Natural Language Processing, provides researchers with a wealth of information on chemical-disease interactions extracted from the vast PubMed database. This publicly available dataset, consisting of over 1500 manually annotated abstracts, serves as a crucial benchmark for developing and evaluating NLP methods focused on information extraction from biomedical texts [36].

In the context of our work, we will focus our analysis on the BioCreative VI ChemProt dataset, recognized as a fundamental resource for studying chemical-protein interactions in scientific literature. This corpus, containing 6,500 interactions spread across 1,820 PubMed abstracts with standardized annotations, offers a diversity of interaction types and

valuable contextual information. We intend to leverage this dataset to deepen our understanding of drug discovery, protein functions, and personalized medicine.

Here's a comparison table 1.1 highlighting the key variations among the datasets we've discussed :

Feature	**BioCreative VI ChemProt**	**BioRED**	**BioInfer**	**BioCreative V CDR**
Entity types	Chemicals, proteins	Genes/proteins, diseases, chemicals, variants, species, cell lines	Varied	Chemicals, diseases
Relation types	4 (binding, positive modulation, negative modulation, Pt modification)	8 (positive regulation, negative regulation, part-of, etc.)	10	Various chemical-disease relations
Size	1820 abstracts	600 abstracts	1100 sentences	1500 abstracts
Relation level	Document level	Document level	Sentence level	Document level
Data source	PubMed abstracts	PubMed abstracts	PubMed abstracts, GENIA corpus, PID	PubMed abstracts

TABLE 1.1 – Benchmark Datasets

1.6 RE Evaluation Metrics

The evaluation metrics for RE models are determined by the extraction approach used in these relations. The evaluation of the performances is based on the use of precision (P), recall (R), and F1-score (F1) models. Understanding these measures requires the use of the confusion matrix.

1.6.1 Confusion Matrix

A confusion matrix is a table that is used to evaluate the performance of a classification algorithm. It provides a detailed breakdown of the model's predictions by comparing them to the actual outcomes. The confusion matrix consists of four entries : true positives, true negatives, false positives, and false negatives, as specified in Table 1.2 :

		Predicted	
		Positive	Negative
Actual	Positive	True Positive	False Negative
	Negative	False Positive	True Negative

TABLE 1.2 – Confusion Matrix.

Using the values from the confusion matrix, we can calculate various evaluation metrics, such as precision, recall, accuracy, and F1 score.

1.6.2 Precision

Precision measures the proportion of correctly predicted positive relations (true positives) out of all predicted positive relations (true positives + false positives). It helps to assess the model's ability to avoid false positive predictions.

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive}$$

1.6.3 Recall

Recall measures the proportion of correctly predicted positive relations (true positives) out of all actual positive relations (true positives + false negatives). It helps assess the model's ability to find all relevant relations.

$$Recall = \frac{TruePositive}{TruePositive + FalseNegatives}$$

1.6.4 F1 Score

The F1 score is the harmonic mean of precision and recall. It provides a single metric that balances both precision and recall.It is particularly useful when dealing with imbalanced datasets or situations where both false positives and false negatives are important. It is often used to compare and rank relation extraction models.The formula for the F1 score is :

$$F1Score = \frac{2 * Precision * Recall}{Precision + Recall}$$

1.6.5 Accuracy

Accuracy is a commonly used evaluation metric for classification tasks, including relation extraction. It measures the proportion of correctly classified instances (true positives and true negatives) out of the total number of instances. The accuracy formula is :

$$Accuracy = \frac{TruePositive + TrueNegative}{TruePositive + TrueNegative + FalsePositive + FalseNegatives}$$

Accuracy provides an overall measure of how well the model is performing across all classes. In such cases, a high accuracy score can be misleading, as the model may be biased towards the majority class. Therefore, it's important to consider accuracy along with other metrics such as precision, recall, and F1 score to obtain a more comprehensive evaluation of the model's performance.

1.7 Conclusion

In this chapter, we explored Relation Extraction. First, we have defined what Relation Extraction is, its importance in the biomedical field, and the motivation behind our research. Then, we have outlined the most important approaches, techniques, and datasets used to model and benchmark RE systems. Finally, we examined the evaluation metrics utilized for gauging the efficacy of RE models.

The next chapter will delve into the utilization of neural networks in Natural Language Processing to enhance Relation Extraction, specifically within the biomedical context.

Chapter 2

Deep Learning and Transformers

2.1 Introduction

Deep learning has emerged as a powerful paradigm within artificial intelligence, revolutionizing various fields by enabling machines to learn complex patterns and representations directly from data. In the realm of Natural Language Processing, deep learning techniques have led to significant advancements in understanding and processing human language.

In this chapter, we delve into the fundamentals of deep learning and explore the evolution of neural network architectures. We begin by exploring the basics of neural networks, including the perceptron and multi-layer perceptron, and their learning algorithms such as backpropagation. We then delve into the application of neural networks in NLP, focusing on recurrent neural networks (RNNs), long short-term memory (LSTM) networks, and their variants, which laid the groundwork for transformer-based models. We discuss the key components of transformer models and their applications in various NLP tasks, including relation extraction.

Furthermore, we examine transfer learning techniques, which have become increasingly prevalent in NLP, allowing models to leverage pre-trained representations and adapt them to specific tasks with limited annotated data. We review prominent transformer-based models such as BERT, RoBERTa, XLNet, and ERNIE, highlighting their architectures and applications in relation extraction from biomedical texts. Finally, we provide an overview of related works in the field, presenting a comparative analysis of various deep learning and transformer-based models for relation extraction.

2.2 Deep Learning

Deep learning is a sub-branch of machine learning which is based on a stack of layers of neural networks. The goal of deep learning is to be able to imitate the actions of the human brain using artificial neural networks. The more layers and neurons the network contains, the more expressive the model will be, which allows it to understand more complex concepts and better adapt to reality.

The beggining of artificial neurons dates back to the **1940s**, when **Warren McCulloch** and **Walter Pitts**[37] proposed their first mathematical and computer model of the biological neuron : **the formal neuron**. This artificial neuron has one or more inputs and a binary output, its operation is simple ; the neuron activates its output (active output = 1) depending on whether its inputs exceed a certain threshold.

The **perceptron** was invented in **1957** by **F. Rosenblatt** [38]. The perceptron is a formal neuron, the smallest possible neural network, whose activation function is a step function also called linear threshold function, which makes the perceptron a linear threshold unit. The perceptron takes arbitrary numbers as inputs (unlike the **formal neurons**), and **each input is weighted by a weight (w)**. The principle of the perceptron is to classify the input data into two groups (0 or 1). However, a perceptron can only classify data that is linearly separable, meaning data that can be separated into two groups.

Multilayer perceptrons are neural networks and aim to classify more complex data than that classified by a perceptron. To do this, the multilayer perceptron observes each of the data it possesses and updates each weight of each neuron in each layer of its network in order to best classify this database. The algorithm that multilayer perceptrons use to update their weights is called error gradient backpropagation.

The revolution of deep learning is linked to the increasing power of computers, enabling the creation and training of neural networks with dozens of hidden layers. The rise of deep learning also stems from the abundance of data continuously accumulating.

2.3 Artificial Neural Networks

Artificial Neural Network is an information processing paradigm that is inspired by the way biological nervous systems, such as the brain, process information. The key ele-

ment of this paradigm is the novel structure of the information processing system. An artificial neural network is composed of numerous interconnected processing elements, akin to neurons, collaborating harmoniously to tackle designated tasks. Similar to human cognition, ANNs acquire knowledge through exposure to examples. Each ANN is tailored for a particular purpose, be it pattern recognition or data classification, achieved through a learning phase. In biological systems, learning encompasses modifications to synaptic connections among neurons [39].

Main Architectures of ANNs

The basic structure of an ANN can be modelled as shown in Figure2.1. This architecture consists of three basic parts [40] :

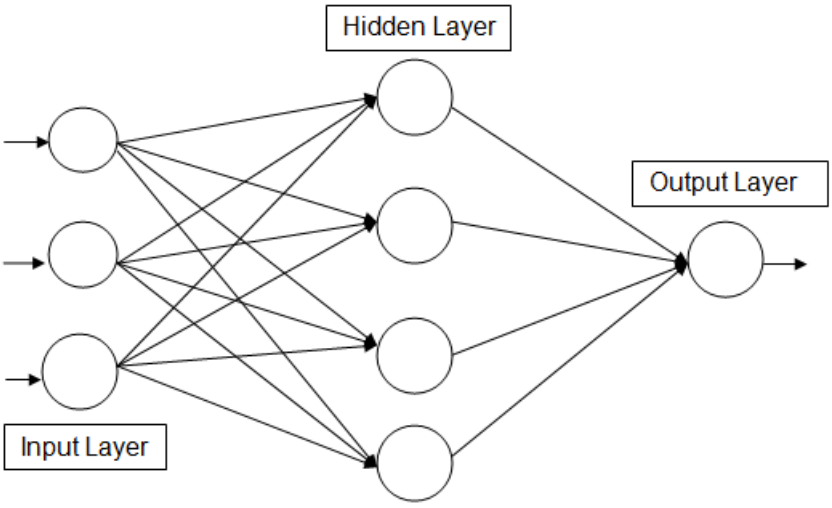


FIGURE 2.1 – Architecture of ANNs.

- **Input Layer :** serves as the initial point for receiving data from the external environment.
- **Hidden, intermediate, or invisible layers :**These layers carry out the fundamental operations within a network, consisting of neurons tasked with extracting features.
- **Output Layer :**This layer, comprised of neurons, generates and delivers the ultimate outputs of the network.

Formal neuron

Formal neurons are elementary units in an artificial neural network. They are a mathematical function. The following diagram represents the general mathematical model of a formal neuron [1] :

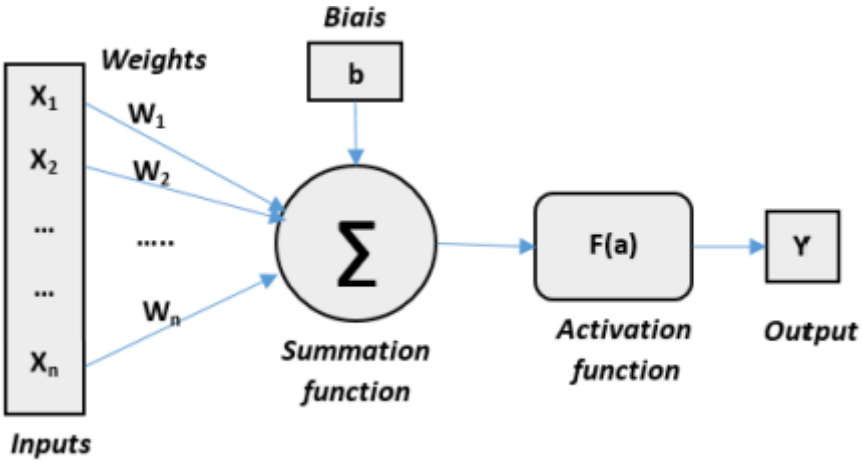


FIGURE 2.2 – Mathematical model of the formal neuron [1].

— The formal neuron that is given in the figure above has n inputs, denoted as : $\{X_1, X_2, \ldots, X_n\}$.

— For each line connecting these inputs to the summation function, a weight is assigned, denoted as : $\{W_1, W_2, \ldots, W_n\}$.

— The summation function aggregates the products of each input multiplied by its corresponding weight, adding the bias term 'b' to adjust the output in conjunction with the weighted sum of inputs to the neuron :

$$Sum = \left(\sum_{i=1}^{n} (x_i \times w_i) + b \right) \tag{2.1}$$

— The activation function F(a) is one of the most important parts of a neuron. Several activation functions can be considered, such as : linear function, sigmoid function, etc. [1]. For example, the sigmoid function transforms the values of the summation into values between 0 and 1. It can be defined as follows :

$$f(x) = \frac{1}{1 + e^{-Sum}} \tag{2.2}$$

— Output :the final activation y :

$$y = f \left(\sum_{i=1}^{n} x_i \cdot w_i + b \right) \tag{2.3}$$

2.4 Neural Network Learning

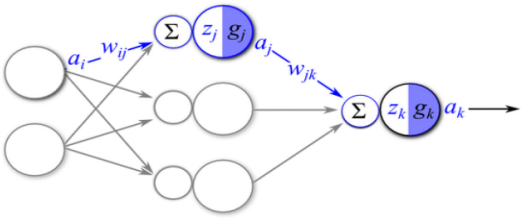
An artificial neural network's ability to learn is arguably the most important aspect. There are two types of neural network learning [41] :

- **Supervised Learning :** This form of learning relies on input data paired with their corresponding correct outputs. During training, the neural network adjusts the weights to achieve optimal values that allow it to generate accurate outputs corresponding to the given inputs. This enables the network to produce correct outputs for any new input. The Backpropagation Algorithm is commonly employed as the training algorithm for supervised learning.
- **Unsupervised Learning :** In contrast, unsupervised learning operates solely on input data without any provided correct outputs. The network's objective is to identify relationships and similarities among the input data, grouping them into distinct categories by extracting patterns unique to each category. As a result, the network can generate output based on the patterns observed in novel input data.

Backpropagation Algorithm

Backpropagation is a key algorithm used to train neural networks, particularly in supervised learning settings. It involves iteratively adjusting the network's parameters (weights and biases) to minimize the difference between predicted and actual outputs. The backpropagation algorithm consists of several steps :

1. **Forward-propagate :** Forward propagation involves sending the inputs through the network layers towards the outputs, as illustrated in Figure 2.3. The output of the network, denoted as a_k, is obtained by applying pre-activation z_l, and activation g_l, for all layers (indexed with i for the input layer, j for the hidden layer, and k for the output layer) using equation 2.4.



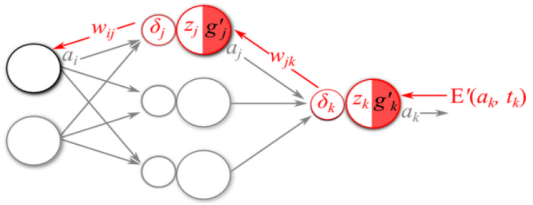
$$a_k = g_k(b_k + \sum_j g_j(b_j + \sum_i a_i w_{ij}) w_{jk}) \tag{2.4}$$

FIGURE 2.3 – Forward-propagate [2].

2. **Back-propagate :** In the second stage of the algorithm, the error E is computed between the network output a_k and the actual output t_k. This calculation is performed using a cost function as described by equation 2.5, which may vary in complexity, ranging from simple options like Mean Squared Error (MSE) to more intricate choices such as cross-entropy.

$$E = \frac{1}{2}\sum_{k \in K}(a_k - t_k)^2 \tag{2.5}$$

The error signal δ' is determined using the following equations (2.6, 2.7) to propagate backward through the network layers, illustrated in Figure 2.4. δ_k represents the error signal for the output layer, while δ_j signifies the error signal for the hidden layer.

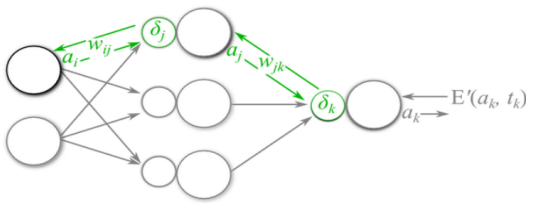


$$\delta_k = g'_k(z_k)E'(a_k, t_k) \tag{2.6}$$

$$\delta_j = g'_j(z_j)\sum_k \delta_k w_{jk} \tag{2.7}$$

FIGURE 2.4 – Back-propagate [2].

3. **Calculate parameter gradient :** The third step involves calculating the gradients of the error function for each layer's weights. This utilizes "forward signals" (denoted as $al-1$) from the preceding layer and "backward error signals" (denoted as δl) from the current layer, as detailed in Equation 2.8. Similarly, the gradient for biases is computed using the same fundamental rule but with a key distinction. Unlike weights, biases are not directly connected to the previous layer, resulting in their "feed-forward activations" always being one layer ahead, as illustrated in Equation 2.9.



$$\frac{\partial E}{\partial w_{l-1,l}} = a_{l-1}\delta_l \tag{2.8}$$

$$\frac{\partial E}{\partial b_l} = b_l\delta \tag{2.9}$$

FIGURE 2.5 – Compute parameter gradients [2].

4. **Update parameters :** The final step entails updating all network parameters - both weights and biases - based on the gradients computed in the previous step. This update utilizes the learning rate parameter (η) as a scaling factor, as specified by the following equations :

$$w_{l-1,l} = w_{l-1,l} - \eta \frac{\partial E}{\partial w_{l-1,l}} \tag{2.10}$$

$$b_l = b_l - \eta \frac{\partial E}{\partial b_l} \tag{2.11}$$

2.5 Neural Networks for NLP

Neural networks have been highly successful in various NLP tasks, including language modeling, machine translation, and sentiment analysis. This section explores different types of neural networks commonly used in NLP, including recurrent neural networks, long short-term memory networks, and transformers.

2.5.1 The recurrent neural networks (RNN)

Recurrent neural networks are a specific type of neural network designed for handling sequential data, such as text. These networks incorporate a feedback loop that aids in retaining information from previous steps. Figure 2.6 illustrates the unrolled representation of a recurrent network. In its unrolled form, it resembles a multi-layer feed-forward network, with the distinction that parameters are shared across the different time steps within the RNN.

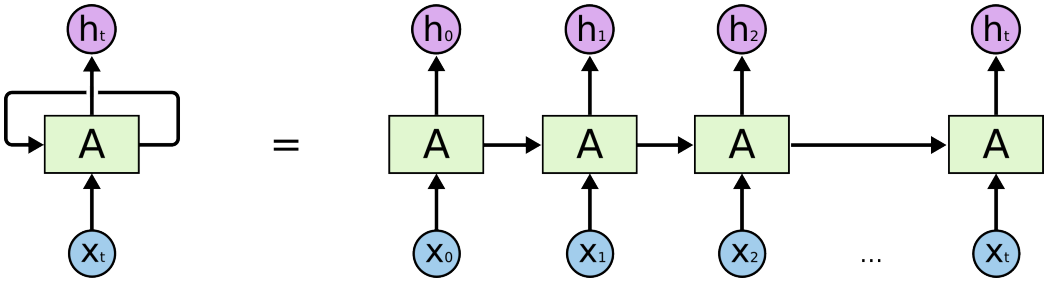


FIGURE 2.6 – Recurrent Neural Network [3].

Among the various versions of recurrent neural networks (RNNs), one widely utilized variant is the one proposed by Elman in 1990 [42].

$$h_t = \rho(W_h x_t + U_h h_{t-1} + b_h) \tag{2.12}$$

$$y_t = \rho(W_y h_t + b_y) \tag{2.13}$$

In the given equation, $\rho(.)$ represents any non-linear activation function, while x_t and h_t represents the input and hidden states at time step t, respectively. The network parameters, including W_h, W_y, U_h, b_h, and b_y, are learned during the training process. The hidden state (h_t) plays a crucial role in retaining past information within the network.

2.5.2 Long Short Term Memory (LSTM)

Hochreiter and Schmidhuber (1997) introduced long short-term memory networks, a specific type of recurrent neural network architecture designed to address the challenges of capturing long-term dependencies and mitigating the vanishing and exploding gradient problems that hinder traditional RNNs [43]. This is achieved by replacing the recurrent hidden layer with a more complex cellular structure, as illustrated in Figure 2.7. The key components of an LSTM cell include :

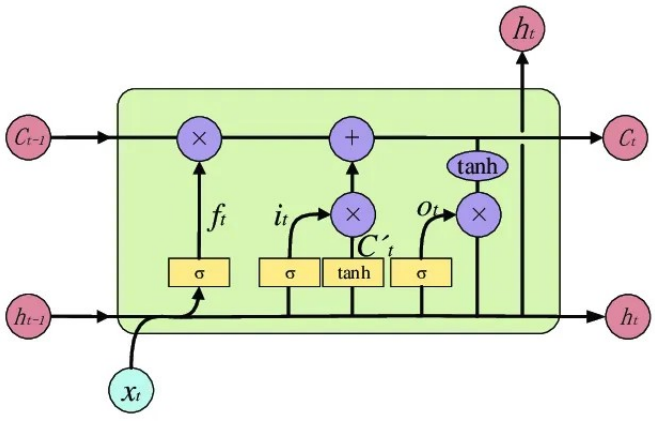


FIGURE 2.7 – LSTM cell [4].

1. **Cell State (C_t) :** This is the core of the LSTM cell, and it's responsible for storing long-term information. It is updated based on the previous cell state (C_{t-1}), the current input (x_t), and the forged gate (f_t).
2. **Gates :** LSTM networks use gating mechanisms to control the flow of information and selectively update the cell state. The three main types of gates in an LSTM are :

- **Input Gate (i_t) :** Determines how much new information to add to the cell state.
- **Forget Gate (f_t) :** Determines how much information to discard from the previous cell state.
- **Output Gate (o_t) :** Determines how much of the cell state should be exposed as the output.

3. **Hidden State (h_t) :** The hidden state contains information selectively filtered from the cell state and serves as the output of the LSTM cell.

LSTMs are able to learn long-term dependencies in sequential data by selectively remembering and forgetting information over time. This makes them well-suited for a variety of tasks, such as speech recognition, machine translation, and time series forecasting.

2.5.3 Transformers

The Transformer is a sequence to sequence type of neural networks introduced by Vaswani et al. (2017) [5] in their paper titled **"Attention is All You Need"**. It differs from Recurrent Neural Networks in two key ways : architecture and processing. Unlike RNNs, which rely on recurrent units that process data sequentially, transformers leverage an attention mechanism alone. This enables them to analyze the entire input sequence simultaneously, a capability that unlocks the power of parallelization during training and inference. This efficiency is a hallmark of the transformer architecture, which elegantly combines a **multi-head self-attention mechanism** with an **encoder-decoder structure**.

The encoder-decoder structure plays a crucial role in how transformers operate. The encoder meticulously creates a compressed vector representation (embedding) of the input sequence, capturing its core essence. The decoder, comprised of stacked layers, builds upon the encoder's output to ultimately generate the final sequence.

2.5.4 Model Architecture

Transformer architecture has revolutionized Natural Language Processing. This architecture departs from its predecessor, the recurrent neural network, by employing a meticulously designed encoder-decoder structure that hinges on the transformative po-

wer of the attention mechanism. The encoder, comprised of stacked layers equipped with self-attention, meticulously analyzes the input sequence (e.g., a sentence) to generate rich contextual representations for each element. This self-attention mechanism empowers the model to selectively focus on relevant portions of the input sequence itself, enabling the capture of long-range dependencies and intricate relationships within the data. The decoder, also utilizing stacked layers with masked self-attention and attention over the encoder's outputs, subsequently generates the final output sequence, such as a translated sentence or a concise summary [5].

Encoder

The encoder depicted in the left half of Figure 2.8 consists of six identical layers, with each layer composed of two sublayers :

1. The first sublayer employs a multi-head self-attention mechanism, where each head receives a linearly projected version of the queries, keys, and values. These heads produce outputs in parallel, which are then combined to generate a final result.
2. The second sublayer consists of a fully connected feed-forward network with Rectified Linear Unit (ReLU) activation. It consists of two linear transformations :

$$\mathrm{FFN}(x) = \mathrm{ReLU}(W_1 x + b_1)W_2 + b_2 \tag{2.14}$$

Each layer applies the same linear transformations to all words in the input sequence, but with different weight (W_1, W_2) and bias (b_1, b_2) parameters. Both sublayers have a residual connection around them. Additionally, each sublayer is followed by a normalization layer, $LayerNorm(x + Sublayer(x))$, which normalizes the sum computed between the sublayer input (x) and the output generated by the sublayer $(Sublayer(x))$.

Decoder

The decoder, visualized on the right side of Figure 2.8, is structured as a stack of six identical layers. Each of these layers contains three sublayers with specific functions :

1. The first sublayer is a masked multi-head self-attention layer, enabling the model to attend to different parts of the input sequence.
2. the second layer uses "multi-head attention" to focus on key aspects of the previous output (encoder stack).

3. The third sublayer is a fully connected feed-forward network that applies non-linear transformations to the outputs of the second sublayer.

Additionally, each of the three sublayer in the decoder has residual connections around them and is followed by a normalization layer.

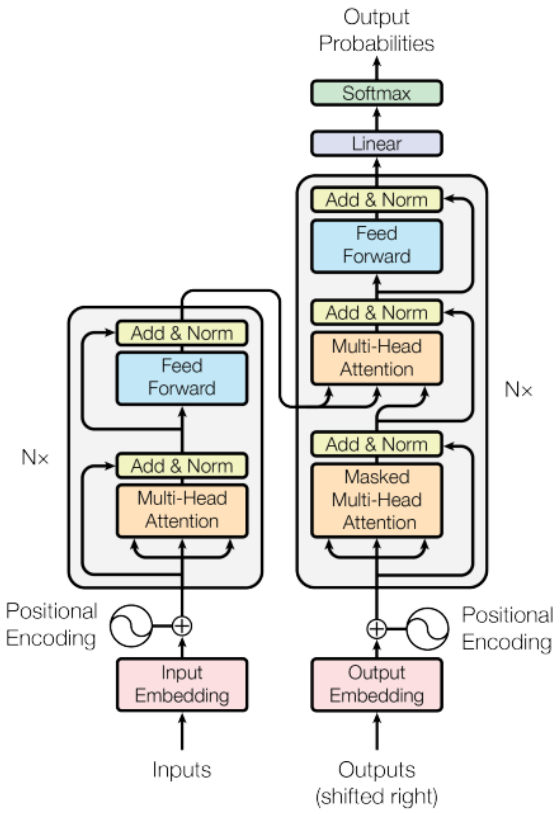


FIGURE 2.8 – Architecture of the Transformer model [5].

Attention

Attention mechanisms utilize vectors to transform a query and a collection of key-value pairs into a single output. The output can be understood as a weighted combination of the values, where each weight is determined by a compatibility function. This function measures how well the query aligns with its corresponding key, essentially gauging their similarity [5].

- **Scaled Dot-Product Attention :**

  Scaled Dot-Product Attention, as depicted in Figure 2.9 (a), is a foundational component of transformer models, enabling precise computation of the relevance of

each token in an input sequence concerning a given query. Mathematically, given a query Q, a set of key vectors K , and a set of value vectors V for an input sequence, the attention scores $Attention(Q, K, V)$ are computed as follows :

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) \tag{2.15}$$

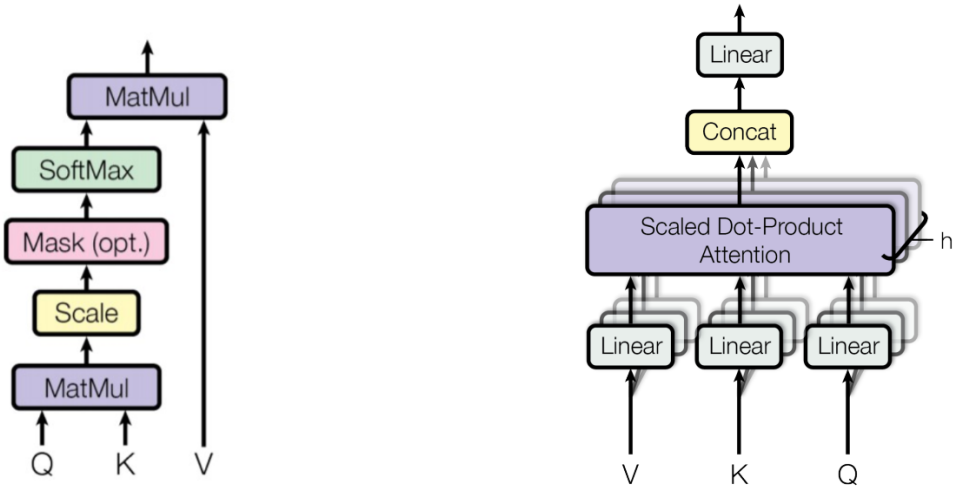
where Q denotes the query, K represents the set of key vectors, V signifies the set of value vectors, and d_k indicates the dimensionality of the key vectors. This mechanism efficiently yields weighted sums of the value vectors, thereby producing the final output [5].

- **Multi-Head Attention :**

Multi-Head Attention, as illustrated in Figure 2.9 (b), enhances the capabilities of scaled dot-product attention by enabling simultaneous attention across multiple subspaces within the input sequence. It achieves this by projecting the query, key, and value vectors into distinct lower-dimensional spaces, or "heads", and performing Scaled Dot-Product Attention independently within each subspace. Given h attention heads, the output of multi-head attention $MultiHead(Q, K, V)$ is computed as follows :

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \text{head}_2, \ldots, \text{head}_h)W^O \tag{2.16}$$

Where each $head_i$ is the result of applying scaled dot-product attention to the projected query, key, and value vectors of the i-th head. W^O is a learnable linear transformation applied to the concatenated outputs. This mechanism enables the model to attend to different aspects of the input sequence simultaneously, enhancing its capacity for nuanced representation learning [5].



(a) Scaled Dot-Product Attention

(b) Multi-Head Attention

FIGURE 2.9 – Attention Mechanisms in Transformers [5].

2.5.5 Transfer Learning

Transfer learning is a method that uses the knowledge acquired from one task to address a related but distinct problem. Essentially, given a source domain Ds and learning task Ts, along with a target domain Dt and learning task Tt (where $Ds \neq Dt$ or $Ts \neq Tt$), transfer learning aims to facilitate the improvement of learning the target predictive function ft in Tt by utilizing the knowledge from Ds and Ts . Figure 2.10 illustrates this process visually. Initially, a mathematical model (referred to as the "Model" along with the Head) is employed to learn the base task Ts. Subsequently, the pre-trained portion of the Model is isolated and combined with a New-Head to create a new model, which then learns the target task Tt. Transfer learning has the potential to enhance the robustness of models and expedite learning in deep learning systems [6].

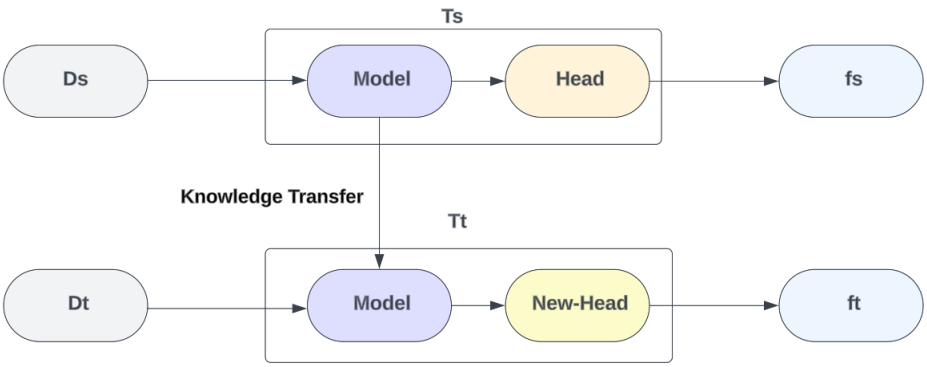


FIGURE 2.10 – Transfer Learning [6].

2.6 Transfer Learning in NLP

Transfer learning has emerged as a powerful technique in natural language processing, allowing models to leverage knowledge from pre-trained representations to improve performance on downstream tasks. In NLP, two common approaches to transfer learning are feature-based transfer and fine-tuning.

Feature-based transfer

Feature-based transfer learning involves extracting relevant features from a pretrained model and using those features as input to a new task-specific model. In NLP, this can be done by using pretrained word embeddings or language models as feature extractors.

Word embeddings, such as Word2Vec [44] and GloVe [45], capture semantic and syntactic relationships between words. These pretrained embeddings can be used as input features for downstream NLP tasks, such as sentiment analysis or text classification. By leveraging the knowledge encoded in these embeddings, the model benefits from the transfer of general language understanding.

Additionally, language models like ELMo [46] and GPT [47] can be used as feature extractors. These models generate contextualized word representations that capture the meaning of words in the context of the surrounding words. These contextualized embeddings can be passed through task-specific models, such as recurrent neural networks or convolutional neural networks, to perform various NLP tasks.

Fine-tuning

Fine-tuning involves taking a pretrained model and updating its parameters on a task-specific dataset. In NLP, fine-tuning is commonly used with models like BERT [48] and other transformer-based architectures.

BERT, for example, is pretrained on a large corpus using unsupervised learning objectives. After pretraining, the model is fine-tuned on a specific downstream task by adding a task-specific layer on top of the pretrained model and training the entire model on task-specific labeled data. During fine-tuning, the pretrained parameters are updated while the task-specific layer is trained from scratch. Fine-tuning allows the model to adapt its representations to the specific task, incorporating task-specific information while retaining the knowledge from the pretrained model.

Fine-tuning has been shown to be highly effective in NLP, achieving state-of-the-art performance on a wide range of tasks, including text classification, question answering, and named entity recognition.

2.7 Transformer-based models in RE

Transformer-based models have significantly advanced relation extraction tasks by capturing contextual information and achieving state-of-the-art results. Here are some popular transformer-based models used in relation extraction :

BERT

In the realm of natural language processing, where machines strive to understand and interact with human language, **the bidirectional encoder representations from transformers (BERT)** model stands as a landmark achievement. Introduced by Google AI in 2018 [48], BERT has revolutionized the field with its exceptional capabilities in capturing nuanced details and contextual information within language.

The cornerstone of BERT lies in its reliance on the **Transformer architecture**. This novel architecture utilizes the **attention mechanism**, allowing it to analyze the relationships between words, regardless of their distance in the sentence. Unlike traditional models that process text sequentially, BERT can simultaneously consider all words in a sentence, enabling it to grasp **long-range dependencies** and understand the context

surrounding each word.

However, the true power of BERT lies in its **pre-training process**. Unlike models trained solely for specific tasks, BERT benefits from extensive pre-training on massive datasets of text. This pre-training involves two crucial steps :

1. **Masked Language Modeling (MLM) :** randomly masking words in the text and predicting the missing words based on the surrounding context. This helps BERT learn the relationships between words and their meanings.
2. **Next Sentence Prediction (NSP) :** Predicting whether two given sentences appear consecutively in the original data. This hones BERT's ability to understand how sentences relate to each other and flow coherently.

Through this pre-training, BERT acquires a deep understanding of general language patterns and contextual information. This knowledge can then be **fine-tuned** for various NLP tasks.

RoBERTa

RoBERTa (Robustly Optimized BERT Pretraining Approach) is an extension of the BERT model that was introduced by Liu et al. (2019)[49]. It addresses some limitations of BERT and incorporates several modifications to improve performance in various natural language processing tasks, including relation extraction. RoBERTa largely follows the architecture and training methodology of BERT but introduces modifications to the training process. Some key modifications include :

1. **Dynamic Masking :**while BERT uses static masking, where a fixed percentage of input tokens are masked during training, RoBERTa applies dynamic masking. This means that the masking pattern is randomly selected for each training instance, allowing the model to see various masked tokens configurations.
2. **Larger Training Corpora :** RoBERTa is trained on a larger amount of data compared to BERT, incorporating additional publicly available text sources. This increased training data helps in capturing a broader range of linguistic patterns and improves the model's generalization ability.
3. **Training Duration :** RoBERTa is trained for a longer duration compared to BERT. The longer training duration allows the model to see more data and learn more effectively.

By incorporating these modifications, RoBERTa achieves improved performance over BERT on a range of NLP tasks, including relation extraction. The enhanced training methodology and larger training corpora allow RoBERTa to learn more robust representations, capturing a broader understanding of language semantics and improving its ability to extract relations between entities in a given sentence.

XLNet

XLNet, introduced by Yang et al. in 2019 [50], stands as a state-of-the-art pre-trained language model that pushes the boundaries of natural language processing performance. Its architecture extends the transformer model, incorporating innovative techniques to address limitations observed in previous models like BERT and GPT.

Key among these innovations is permutation language modeling, a departure from BERT's token masking approach. XLNet considers all permutations of the input sequence during training, allowing it to capture bidirectional context more effectively. To efficiently handle this task, XLNet utilizes an autoregressive factorization scheme, breaking down the joint probability of permutations into conditional probabilities. Additionally, XLNet employs a two-stream self-attention mechanism, which captures information from both preceding and succeeding tokens, enhancing its ability to model bidirectional dependencies within the input sequence.

ERNIE

ERNIE (Enhanced Representation through kNowledge IntEgration) is a transformer-based model specifically designed to enhance representation learning in natural language processing tasks, including relation extraction. It was introduced by Sun et al. (2019) [51].

The key idea behind ERNIE is to incorporate external knowledge sources, such as knowledge graphs or other structured knowledge bases, into the pretraining process. By integrating external knowledge, ERNIE aims to improve the model's understanding of relations between entities and enhance its ability to extract useful information from text.

ERNIE's training process involves two steps : pretraining and fine-tuning.

In the **pretraining phase**, ERNIE uses a masked language modeling objective similar to BERT. However, ERNIE introduces additional training objectives that utilize external knowledge. For example, ERNIE incorporates entity-level and sentence-level knowledge

objectives, where the model is trained to predict masked entities and their relationships based on the knowledge in external sources.

In the **fine-tuning phase**, ERNIE is further trained on task-specific labeled data, such as relation extraction datasets. The model's parameters are adjusted to make accurate predictions based on the specific task requirements. Fine-tuning allows ERNIE to adapt its pretrained representations to the relation extraction task and improve its performance in extracting relations between entities.

By integrating external knowledge sources during both pretraining and fine-tuning, ERNIE aims to provide enhanced representations that capture both contextual information and knowledge-based signals. This integration of knowledge helps ERNIE achieve improved performance in relation extraction tasks by leveraging the rich information available in external knowledge bases.

2.8 Related Works

This section presents a survey of existing research efforts in the field of relation extraction within the biomedical domain, focusing on research conducted using the ChemProt dataset and utilization of deep learning techniques.

SciBERT model, developed by **Beltagy et al. [27]** , represents a significant breakthrough in natural language processing tailored specifically for scientific text. By leveraging the BERT architecture and training on a vast corpus of scientific literature from sources like pubMed and arXiv, SciBERT has acquired a deep understanding of the nuances and complexities inherent in scientific language. When evaluated on the ChemProt relation extraction benchmark, SciBERT achieves an F1-score of 83.64. This F1-score signifies SciBERT's capability in identifying relationships between chemical and protein entities within a scientific document.

Lee et al. [13] introduced BioBERT, a pre-trained biomedical language representation model tailored explicitly for biomedical text mining tasks. BioBERT builds upon the BERT architecture but is pre-trained on PubMed abstracts and pubMed central full-text articles, allowing it to capture domain-specific biomedical knowledge and context effectively.In relation extraction on ChemProt, Lee et al. demonstrate BioBERT's efficacy,

achieving an F1 score of 76.46 in accurately discerning relationships between chemicals and proteins within the ChemProt dataset.

In their work, **Alrowili et al.[52]** recognized the limitations of single-architecture models for biomedical relation extraction tasks like those found in the ChemProt benchmark. They proposed BioM-BERT, a powerful pre-trained language model that leverages a multi-model ensemble approach. This approach combines the strengths of three prominent architectures – BERT, ALBERT, and ELECTRA – all fine-tuned on a massive corpus of biomedical text. This strategy allows BioM-BERT to capture the nuances of scientific language and the relationships between chemical entities and proteins. Notably, the authors report that BioM-BERT achieves an F1 score of 80.0 on the ChemProt task, demonstrating its effectiveness in accurately identifying chemical-protein interactions.

Shin et al. [53] proposed BioMegatron, a foundational large language model (LLM) specifically designed for the biomedical domain. It's training on a massive biomedical text corpus equips it to understand scientific language crucial for ChemProt relation extraction. Achieving an F1-score of 77.0 on the benchmark demonstrates BioMegatron's capability in this task. Future research on fine-tuning and specialized architectures holds promise for further improvement. BioMegatron offers a valuable tool for researchers to unveil chemical-protein interactions from biomedical literature.

Yasunaga et al. [54] proposed LinkBERT, a document-centric pre-trained language model (PLM) that leverages document-level links for training. BioLinkBERT (large), a large-scale biomedical variant of LinkBERT, achieves competitive micro F1-score and F1-score of 79.98 on the ChemProt benchmark. This focus on document-level connections suggests particular promise for ChemProt tasks, where understanding relationships across scientific text sections is crucial.

2.8.1 Comparative table

Table 2.1 provides a summary of the previously discussed related works and contrasts them with our proposed approach across various comparison criteria.

Work	**Model**	**Training Corpus**	**Task Fine-tuning**	**Evaluation Dataset**	**Score [F1]**
Beltagy et al.[27]	SciBERT	Semantic Scholar	Yes	ChemProt	83.64
Lee et al. [13]	BioBERT	PubMed	Yes	ChemProt	76.46
Alrowili et al.[52]	BioM-BERT	PubMed + PMC	Yes	ChemProt	80.0
Shin et al. [53]	BioMegatron	PubMed + PMC	No	ChemProt	77.0
Yasunaga et al.[54]	BioLinkBERT	Wikipedia + PubMed	Yes	ChemProt	79.98

TABLE 2.1 – Comparative table of related works.

2.9 Conclusion

This chapter has provided an overview of deep learning and neural networks, focusing on prominent architectures of Artificial Neural Networks. It discusses neural network learning, particularly with the backpropagation algorithm.

Furthermore, it explores the application of neural networks in natural language processing, including RNNs, LSTMs, transformers, and their model architectures and attention mechanisms. Transfer Learning is also discussed, highlighting its role in utilizing pre-trained models to enhance performance in new tasks, especially in NLP and relation extraction using transformer-based models like BERT, RoBERTa, and XLNet.

Lastly, related works in this field are reviewed, demonstrating the significant impact of DL and transformer architectures, particularly in RE tasks. The next chapter delves into the proposed relation extraction model, leveraging the strengths of deep learning and transformer architectures.

Chapter 3

Proposed Biomedical RE Model

3.1 Introduction

In this chapter, we will discuss our proposed approach for relation extraction in the biomedical domain, leveraging the transformative power of transformers. This solution aims to equip models with the capability to extract chemical-protein relationships from biomedical texts.

The system architecture of the proposed RE model includes multiple components and processes. It leverages the ChemProt dataset as the basis for training and evaluating the model. This dataset features a wide variety of interaction types, enabling a thorough evaluation of the RE model's performance.

Furthermore, the chapter explores the strategic integration of SciBERT, a pre-trained language model, into the proposed RE model's development. This pre-trained model have garnered considerable attention due to their exceptional performance in NLP tasks.

The chapter meticulously describes the crucial preprocessing stage, emphasizing its role in preparing data for the training process. This stage entails tokenization, a process that fragments the text into smaller, more manageable units for analysis.

Finally, the chapter sheds light on the fine-tuning process, a critical step that involves configuring the model architecture and subsequently training it on the prepared data. Training encompasses optimizing the model's internal parameters and meticulously adjusting hyper-parameters to achieve optimal performance.

3.2 System Architecture

Relation extraction with transformer-based models like SciBERT involves a series of crucial steps to deliver accurate responses to user queries. Figure 3.1 depicts the overall RE process.

Preprocessing : Sentences or texts are tokenized into individual tokens, and special tokens like [CLS] and [SEP] are added. Additional markers may be used to highlight the entities involved in the relation. For example, in the sentence "The kinase phosphorylates the substrate," the entities "kinase" and "substrate" might be marked as `[@CHEMICAL$]` and `[@GENE$]`.

Embedding : The tokenized sequences are converted into embeddings, which represent the tokens in a high-dimensional vector space. These embeddings capture the contextual meaning of the tokens.

Encoding : The embeddings are fed into a transformer encoder, which consists of multiple layers that process the tokens and capture their contextual representations. The encoder uses mechanisms like self-attention to dynamically weigh the importance of each token relative to others in the context.

Attention : The encoded tokens undergo processing via a self-attention mechanism, specifically the multi-head attention mechanism. This calculates attention weights for each token, determining its significance relative to other tokens and enriching each token's representation with contextual information from the entire sequence.

Relation Prediction : The output from the attention mechanism is passed through a task-specific layer, typically including linear layers, followed by a soft max activation. The model predicts the type of relationship between the entities based on their contextual embeddings.

Post-processing : The predicted relation type is extracted along with the entities involved. The relation is cleaned and processed for clarity and accuracy, then returned in a structured format such as a triplet (`@CHEMICAL&`, Relation, `@GENE&`).

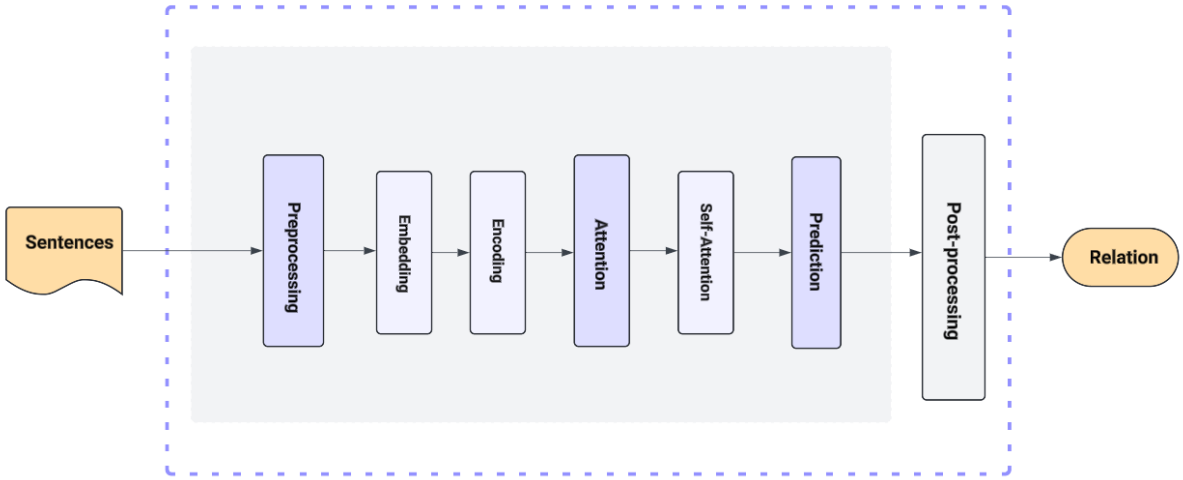


FIGURE 3.1 – Relation Extraction Process Using Transformer-Based Models.

3.3 Used Dataset

We fine-tuned our model on the ChemProt dataset to focus on understanding chemical-protein interactions.

3.3.1 BioCreative VI ChemProt

The ChemProt corpus, employed within the BioCreative VI text mining task [7], encompasses 1,820 PubMed abstracts meticulously annotated by domain experts to explicitly capture chemical-protein interactions. These interactions are noteworthy for their unidirectional nature, focusing exclusively on the influence of chemicals on genes/proteins (chemical-to-gene/protein direction). The specific interaction types encompassed within the corpus are comprehensively categorized in Table 3.1

Group	**Eval.**	**CHEMPROT relations belonging to this group**
CPR :1	N	PART_OF
CPR :2	N	REGULATOR|DIRECT_REGULATOR|INDIRECT_REGULATOR
CPR :3	Y	UPREGULATOR|ACTIVATOR|INDIRECT_UPREGULATOR
CPR :4	Y	DOWNREGULATOR|INHIBITOR|INDIRECT_DOWNREGULATOR
CPR :5	Y	AGONIST|AGONIST-ACTIVATOR|AGONIST-INHIBITOR
CPR :6	Y	ANTAGONIST
CPR :7	N	MODULATOR|MODULATOR-ACTIVATOR|MODULATOR-INHIBITOR
CPR :8	N	COFACTOR
CPR :9	Y	SUBSTRATE|PRODUCT_OF|SUBSTRATE_PRODUCT_OF
CPR :10	N	NOT

TABLE 3.1 – CHEMPROT Relations by Group[7].

The provided table categorizes CHEMPROT relations into distinct groups, each identified by a unique CPR (Chemical-Protein Relation) number, ranging from CPR :1 to CPR :10. It includes two main columns : one indicating whether the group has been evaluated (Eval.) and another listing the specific CHEMPROT relations within each group.

The **Group** column identifies the CHEMPROT relation groups with labels such as CPR :1, CPR :2, etc. The **Eval**. column indicates the evaluation status of each group, where **Y (Yes)**denotes that the group has been evaluated, meaning the relations within this group have been reviewed and validated, and **N (No)** signifies that the group has not been evaluated, potentially indicating either a lack of validation or lower priority for evaluation.

CHEMPROT relations belonging to this group :

- **PART_OF** : Indicates that a chemical is part of a protein or protein complex.
- **REGULATOR, DIRECT_REGULATOR, INDIRECT_REGULATOR** : Indicate relations where the chemical regulates the protein directly or indirectly.
- **UPREGULATOR, ACTIVATOR, INDIRECT_UPREGULATOR** : Indicate relations where the chemical increases the activity of the protein, either directly (activator) or indirectly (indirect_upregulator).
- **DOWNREGULATOR, INHIBITOR, INDIRECT_DOWNREGULATOR** : Indicate relations where the chemical decreases the activity of the protein, either

directly (inhibitor) or indirectly (indirect_downregulator).

- **AGONIST, AGONIST-ACTIVATOR, AGONIST-INHIBITOR** : Indicate relations where the chemical acts as an agonist, activating or inhibiting the protein.
- **ANTAGONIST** : Indicates that the chemical acts as an antagonist, inhibiting the action of the protein.
- **MODULATOR, MODULATOR-ACTIVATOR, MODULATOR-INHIBITOR** : Indicate relations where the chemical modulates the activity of the protein, either activating or inhibiting it.
- **COFACTOR** : Indicates that the chemical acts as a cofactor necessary for the protein's activity.
- **SUBSTRATE, PRODUCT_OF, SUBSTRATE_PRODUCT_OF** : Indicate relations where the chemical is a substrate of the protein, a product of the protein, or both.
- **NOT** : Indicates a lack of a specific relationship between the chemical and the protein.

3.3.2 Data Exploration

A comprehensive analysis and exploration of the dataset proves to be crucial in both the data pre-processing stage and the selection of optimal hyperparameters for our models. These factors have been demonstrably influential in determining the model's training performance.

Format

The ChemProt dataset isprovided in tabular format (such as CSV or TSV files). These files typically contain the following columns :

- Chemical : Unique identifier of the chemical (often a ChEMBL or PubChem identifier).

  "ChEMBL : Focuses on small molecules with bioactivity data ; identified by ChEMBL IDs".

  "PubChem : Provides a comprehensive resource for chemical substances and their biological".

- Protein : Unique identifier of the protein (often a UniProt identifier).
  "UniProt : Offers extensive protein sequence and functional information ; identified by UniProt IDs".
- Interaction Type : Type of interaction between the chemical and the protein, usually categorized into several functional categories (activation, inhibition, binding, etc.).
- Document ID : Identifier of the scientific document from which the information is extracted (often a PubMed ID).
- Sentence : Exact sentence extracted from the scientific literature describing the interaction.
- Relation Type : Type of relation (e.g., direct or indirect).

Chemical	**Protein**	**Interaction Type**	**Document ID**	**Sentence**	**Relation Type**
CHEMBL25	P12345	activation	12345678	Chemical X activates protein Y.	direct
CHEMBL32	Q67890	inhibition	87654321	Chemical A inhibits protein B.	indirect
CHEMBL45	O12345	binding	23456789	Chemical M binds to protein N.	direct

TABLE 3.2 – Chemical-Protein Interactions.

In our study, several modifications have been made to the ChemProt dataset to tailor it for specific research needs. The original ChemProt dataset, which contains comprehensive information on chemical-protein interactions, served as a foundation. However, to better align with research objectives, the focus was placed on refining the data structure and enhancing the clarity of interaction types. Relevant interactions were meticulously extracted, the data was reformatted, and specific labels were introduced to streamline the dataset for analytical models.

Steps of Transformation

– The dataset included columns such as **"Chemical"**, **"Protein"**, **"Interaction Type"**, **"Document ID"**, **"Sentence"**, and **"Relation Type"**. The data struc-

ture was simplified, retaining essential interaction details in the columns **"index"**, **"sentence"**, and **'label'**.

- Sentences were directly extracted from scientific literature, often mentioning specific chemical and protein names. These sentences were standardized by replacing actual chemical and protein names with placeholders `'@CHEMICAL$'` and `'@GENE$'`. This anonymization maintained consistency and focused on interaction patterns rather than specific entities.
- Each interaction was referenced by a **"Document ID"**. A unique indexing system for each interaction in the format **"Document ID"** was introduced. This identifier allowed tracing interactions back to their original scientific document while providing a clear reference system within the modified dataset.
- The dataset contained a **"Relation Type"** column with values such as direct or indirect.A **"label"** column was introduced to categorize interactions. This column includes specific labels like **"CPR :4"** for certain interaction types, while **'false'** indicates non-interactions. This allowed for more detailed and precise classification of interaction.
- A detailed data structure with multiple columns describing each interaction was streamlined to focus on interaction sentences and their corresponding labels. The columns were reduced to **"index"**, **"sentence"**, and **"label"** to make the dataset more manageable and suitable for machine learning models or further analytical processes.
- Natural language processing techniques were used to identify chemical and protein entities in the sentences. Once the entities were identified, they were replaced with standardized placeholders `'@CHEMICAL$'` for chemicals and `'@GENE$'` for proteins). **For example**, a sentence like "Chemical X activates protein Y." was transformed into "@CHEMICAL$ X activates @GENE$."

Example of Transformation

The table 3.3 below illustrates the dataset after the transformation process :

Index	Sentence	Label
1712335.T23.T45	'@CHEMICAL$' (0.3 mg/kg, iv) produced a significant inhibition of the '@GENE$' mediated positive chronotropic response to isoproterenol.	CPR :4
10064839.T47.T55	We found that DF, '@CHEMICAL$', and DR were relative high-affinity ligands at '@GENE$' (Ki=151, 205, 144 nM, respectively) while all of them were with low affinity at sigma-2 receptors (Ki=4-11 microM).	false

TABLE 3.3 – Example of Transformation ChemProt.

Characteristics

The pie chart 3.2 illustrates the distribution of classes in the ChemProt dataset. The largest segment, labeled "false," comprises 79.2% of the data, indicating that a significant majority of the samples do not have an annotated chemical-protein relationship or belong to a non-relevant class. among the annotated relationships, the "CPR :4" class is the most frequent, accounting for 10.4% of the data. Other classes such as "CPR :3" (4.1%), "CPR :9" (3.8%), "CPR :6" (1.5%), and "CPR :5" (1.0%) have much smaller proportions.

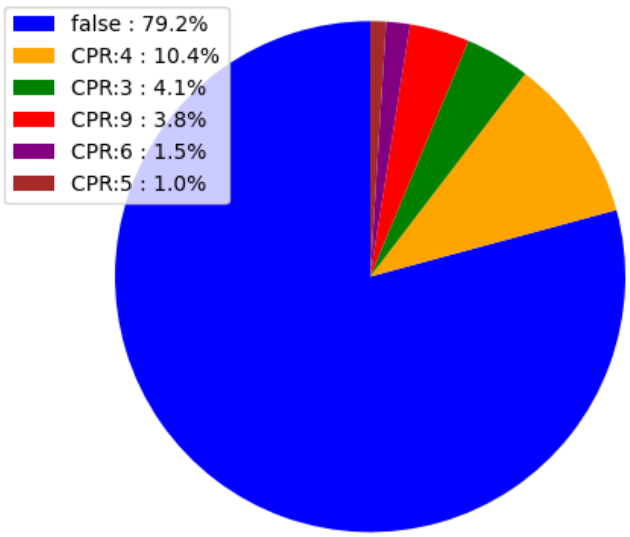


FIGURE 3.2 – Number of examples for each class.

Context Length

The graph 3.3 illustrates the distribution of sentence lengths in the ChemProt dataset.

We observe that the majority of sentences range from 10 to 40 words in length. specifically, the peak of the distribution is around 20 words per sentence, indicating that sentences of this length are the most frequent in the dataset. Beyond 40 words, the number of examples gradually decreases, showing that longer sentences are increasingly less common. Additionally, very few sentences exceed 80 words, and it is extremely rare to find sentences longer than 100 words. This distribution is typical of many text corpora, where shorter and medium-length sentences are more prevalent than very long sentences. For natural language processing models, this information is crucial, as it helps determine preprocessing strategies such as sentence segmentation or handling variable-length sequences.

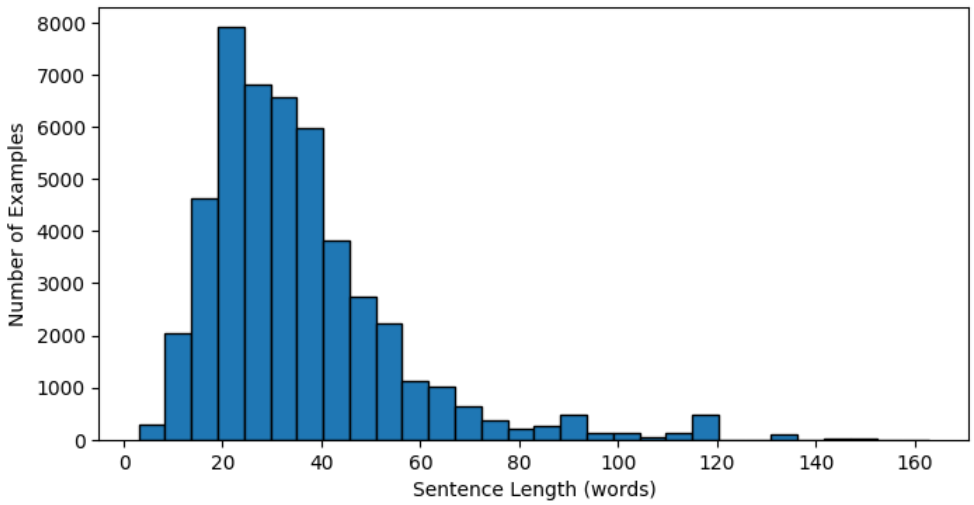


FIGURE 3.3 – Distribution of Context Length.

3.4 Building up the Proposed RE Model

3.4.1 Language Models

SciBERT

SciBERT [27] is a specialized version of the BERT model designed specifically for scientific text. The model's architecture is based on the original BERT, with the key parameters such as the number of layers, hidden size, and the number of self-attention heads remaining the same.

SciBERT is pre-trained on the Semantic Scholar (S2) corpus, which includes 1.14 million papers from computer science and biomedicine, encompassing both abstracts and full papers. This domain-specific corpus allows SciBERT to better capture the nuances and specialized terminology of scientific literature. Similar to BERT, SciBERT comes in versions akin to BERT Base and BERT Large, with SciBERT Base being the most commonly used, featuring 12 layers, 768 hidden units, and 12 self-attention heads per layer.

SciBERT is particularly well-suited for relation extraction, a critical task in natural language processing that involves identifying relationships between entities such as genes, proteins, diseases, and chemicals in scientific texts. A SciBERT-based relation extraction system typically comprises three primary components : input encoding, contextual encoding, and relation classification.

In the input encoding component, the text containing potential relations is tokenized into numerical representations using SciBERT's specialized vocabulary. **Token embeddings**[1] are created for each token, along with **segment embeddings**[2] to differentiate between different parts of the input if necessary, and positional embeddings to capture the sequence order of the tokens.

The contextual embeddings are then used to classify the relationships between entities in the text. This involves using attention mechanisms and fully connected layers to focus on the relevant parts of the context that indicate a relationship.

In the contextual encoding component, SciBERT generates contextual embeddings by

1. Are numerical vectors that represent the meaning of each token, taking into account its position in the sentence.
2. A number is added to each token, indicating its sentence (first or second).

processing the token embeddings through multiple layers of bidirectional Transformers. The model leverages its pre-trained knowledge of scientific texts to understand the context and relationships between words more effectively, providing a rich contextual representation that captures the intricate relationships and dependencies between entities within the scientific text.

Hyper-parameters

The details of the hyperparameters used to fine-tune the pre-trained SciBERT model on the ChemProt dataset are provided in Table 3.4.

The **"Max_length"** is a hyperparameter that acts as a ceiling on the number of tokens a model can process at once. Any sequences exceeding this limit are trimmed down to fit within the specified length.

The **"batch_size"** hyperparameter dictates the number of samples processed together during each training iteration. We chose a specific batch size, which basically means we picked a certain number of examples to process together before the model updates its internal workings.

We employed the **Adam optimizer** for training, incorporating **weight decay** and a specific **learning rate**. Weight decay acts as a regularized, preventing the model's parameters from becoming excessively large. The learning rate, on the other hand, dictates the magnitude of adjustments made to the model's weights during training.

To maintain consistency, all other hyperparameters remained at their default settings, leveraging the values pre-defined within the chosen model architecture.

Hyperparameter	**Value**
max_length	512
train_batch_size	3
learning Rate	2e-05
num_train_epochs	2
weight_decay	0.01

TABLE 3.4 – Hyperparameters for our proposed fine-tuned SciBERT model.

3.4.2 Preprocessing

To prepare text data for our model, we subject it to a preprocessing pipeline. This crucial step transforms the text into a format the model can readily comprehend and utilize. The preprocessing pipeline typically encompasses the following steps :

Entity masking

Entity masking involves replacing specific entities within the text with special tokens to generalize the information they represent. This process is crucial for anonymizing or standardizing the input text, making it easier for us to process and analyze.

In biomedical text, we replace gene mentions with a special token like "@GENE$" to anonymize the specific gene names. For example, if the text mentions the gene "BRCA1", we would replace it with "@GENE$". We also replace chemical mentions with a special token such as "@CHEMICAL$" to standardize their representation. For instance, if the text mentions "aspirin", we would replace it with "@CHEMICAL$".

Tokenization

The tokenizer performs several crucial tasks.Firstly, it breaks down sentences into tokens using the WordPiece method, which effectively manages complex biomedical terminology by decomposing unknown words into subword units. Then, we convert these tokens into unique integer IDs based on SciBERT's extensive vocabulary, transforming the textual data into a numerical format suitable for model ingestion.

For this purpose, we use the huggingface transformer tokenizer, ensuring compatibility with the model architecture we intend to use.

To ensure uniform input lengths, we truncate sentences that exceed the model's maximum length, preserving only the most relevant parts of the text. This process is controlled by the max_length hyper-parameter. For shorter sentences, we apply padding, adding [PAD] tokens until each sequence reaches the desired length.

Additionally, we generate attention masks, which assign a value of 1 to actual tokens and 0 to padding tokens, guiding the model to focus on the substantive content.

Moreover, special tokens such as [CLS] (classification token) and [SEP] (separator token) are added to the tokenized sequences. The [CLS] token is typically added at the beginning of each sequence and is used for classification tasks, while the [SEP] token is

used to separate different segments of text, especially useful in tasks involving pairs of sentences or passages.

3.4.3 Fine-tuning

To tackle our relation extraction task, we leverage the **AutoModelForSequence-Classification** class from the Hugging Face library. Similar to the tokenizer, this class employs the **from_pretrained** method to seamlessly download and cache the pre-trained model. Beyond the model and tokenizer, fine-tuning necessitates an optimizer. We'll employ the **AdamW** optimizer from pytorch, known for incorporating gradient bias correction and weight decay for enhanced training stability. Configuring the optimizer involves specifying the **learning rate** and feeding the model's parameters.

Model training is achieved through the **train** method, iterating over the training data for a predefined number of epochs (complete passes). Following each epoch, the model's performance is assessed on the validation set. This evaluation guides the training process, determining if further iterations are necessary for optimal performance.

3.4.4 Evaluation

To evaluate our model, logits play a crucial role in determining the confidence of the model's predictions. Logits are the raw, unnormalized scores output by the model's final layer before applying an activation function such as softmax. These scores indicate the likelihood of each possible class—in this case, the potential relation types between entities.

For each entity pair in the input data, the model generates logits corresponding to various relation types. Higher logits indicate higher confidence in the associated relation type. To predict the relation between a pair of entities, we typically select the relation type with the highest logit.

Once the logits are obtained, we can apply a softmax function to convert them into probabilities, providing a more interpretable measure of confidence for each relation type. The predicted relation type is then the one with the highest probability.

In the evaluation process, we use these logits to make predictions and compare them against the ground truth. If the highest logits correspond to valid relation types and align well with the ground truth data, it indicates that the model is making accurate predictions.

Filtering and validation steps ensure that the predicted relations are reasonable and adhere to predefined criteria, such as valid entity positions and relation types.

Finally, we calculate evaluation metrics such as precision, recall, and F1 score based on the comparison between the model's predictions and the ground truth relations.

3.5 Conclusion

This chapter introduced the proposed approach for relation extraction in the biomedical domain using transformer architectures. The system architecture of the RE model was detailed, emphasizing the use of the SciBERT language model.

Furthermore, the chapter comprehensively explored the model building process, encompassing data exploration, preprocessing techniques such as tokenization, and the subsequent fine-tuning and training procedures. Finally, the chapter culminated in the evaluation of the proposed RE model, demonstrably showcasing its performance.

The following chapter dives deep into a thorough analysis and discussion of the evaluation results for our proposed relation extraction model. We will also shed light on the challenges and hurdles encountered during the development and implementation stages.

Chapter 4

Experimental Results and Discussion

4.1 Introduction

This chapter delves into the core findings of our research by presenting the experimental results and offering a thorough discussion of these outcomes. It is structured to provide a comprehensive analysis of the performance of our proposed model.

We begin by detailing the experimental setup, including the hardware and software environments utilized for our tests. Following this, we present the results obtained from our model, including performance metrics such as precision, recall, and F1-score. These metrics are critical for evaluating the effectiveness of our model and for comparing it against existing benchmarks.

The discussion section offers insights into the implications of our results, highlighting the strengths and limitations of the model. Key areas of analysis include the confusion matrix, which helps identify specific challenges in relation extraction. We also discuss the model's performance across different types of chemical-protein relations (CPRs).

Finally, we test the proposed RE model to demonstrate its practical applicability.

4.2 Experimental Setup

Table 4.1 details the experimental setup employed in this study. It comprehensively outlines both the hardware and software components that formed the foundation for our rigorous testing and evaluation of the proposed model.

Hardware and Training	**Cloud Tools**	**Google Colaboratory (Colab), by Google :** • CPU : Intel(R) Xeon(R) • GPU : Tesla T4, 16GB • RAM : 12.7GB • Disk : 107.7GB
		Kaggle Notebooks, by Kaggle : • CPU : Intel(R) Xeon(R) • GPU : Tesla P100-PCIE-16GB • RAM : 13GB • Disk : 107.37GB
Software and Libraries	**Programming Language**	Python 3
	Libraries	**Hugging Face Datasets :** a python library for loading and preprocessing datasets. It offers a straightforward and unified interface for loading datasets from various sources, including CSV files, JSON files, and HDF5 files. Additionally, it provides a range of preprocessing functions, such as tokenization, normalization, and filtering.
		Hugging Face TokenizersFast :a state-of-the-art tokenizers library optimized for both research and production. It implements the most commonly used tokenizers in Transformers, prioritizing performance and versatility.
		Hugging Face Transformers : a widely used open-source Python library for NLP tasks. It offers numerous pre-trained Transformer models and a framework for fine-tuning these models on custom tasks.
		PyTorch : an open-source machine learning framework built on the Torch library and utilizing Python. It is employed for a wide range of tasks, including natural language processing, computer vision, and robotics. PyTorch is renowned for its flexibility and ease of use.

TABLE 4.1 – The experimental setup used.

4.3 Results and Discussion

We now present the performance results obtained from our model. Table 4.2 details these findings. The evaluation of our model leveraged standard metrics, including F1 score, precision, and recall.

Figures 4.1 and 4.2 visualize the training and validation accuracy and loss curves for the proposed SciBERT model. These curves offer valuable insights into the model's performance on the training data and its ability to generalize to unseen data.

Model	Precision	Recall	F1 Score
RE-SciBERT	77.15	75.73	90.10

TABLE 4.2 – Performance results of proposed the model.

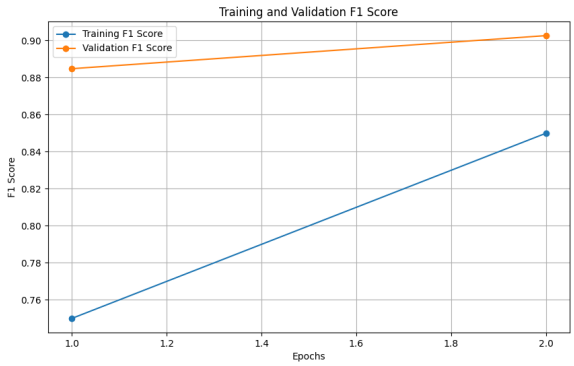


FIGURE 4.1 – F1 Score curve of the proposed model.

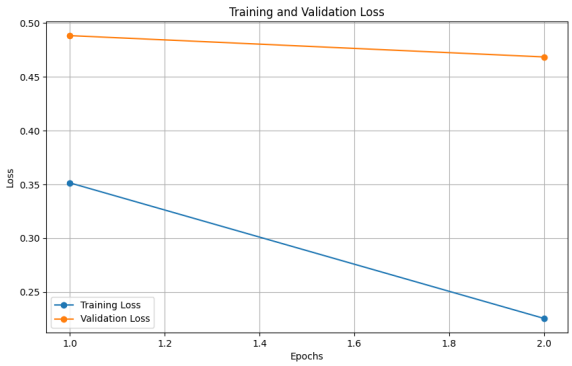


FIGURE 4.2 – Loss curve of the proposed model.

As shown in Figure 4.3, the comparative performance metrics for BioBERT, BioMegatron, and RE-SciBERT models illustrate distinct differences in their effectiveness.

RE-SciBERT stands out with the highest F1 Score of 90.10, indicating its superior balance between Precision and Recall, which are 77.15 and 75.73, respectively.

BioBERT, while slightly behind RE-SciBERT, still performs robustly with a Precision of 77.02, Recall of 75.9, and an F1 Score of 76.46. In contrast, BioMegatron exhibits the lowest performance, with a Precision of 74.5, Recall of 79.7, and an F1 Score of 77, reflecting its comparatively weaker performance.

Overall, RE-SciBERT demonstrates the best performance, particularly in its F1 Score, while BioBERT remains a strong contender, and BioMegatron lags in all evaluated metrics.

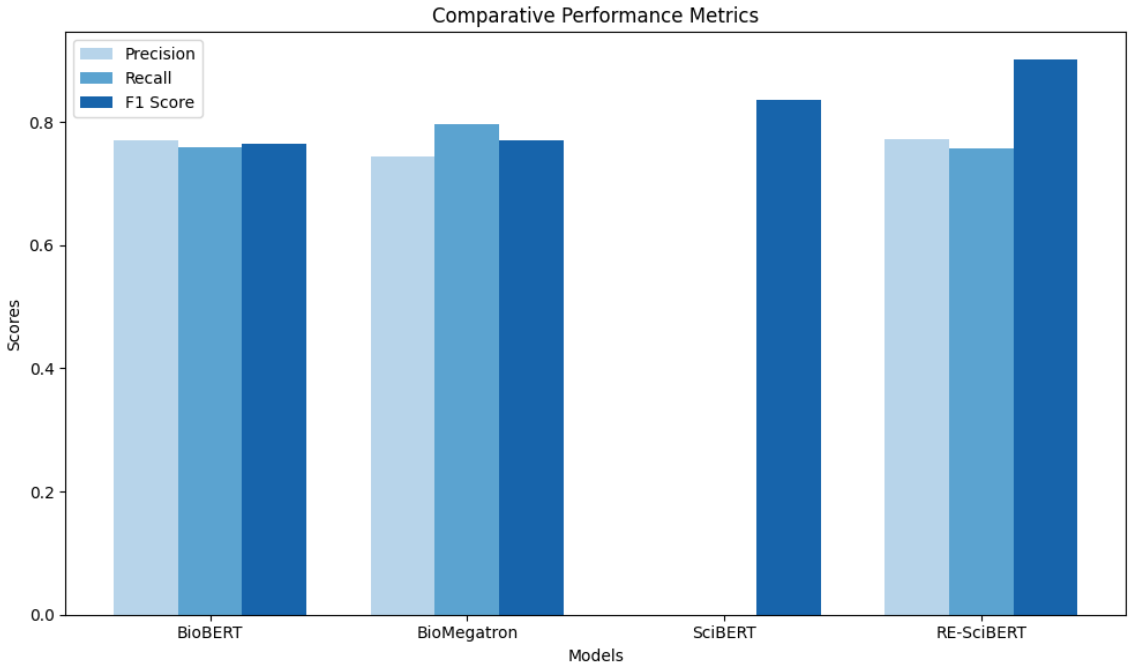


FIGURE 4.3 – Performance Comparison of Models.

Figure 4.4 depicts the F1 score's progression during the training process. As observed, the metric exhibits a gradual increase in the initial two epochs, signifying the model's successful learning and performance improvement.

However, the scores plateau beyond this point, indicating that the model has likely extracted most of the valuable knowledge from the training data. It's important to reiterate, as mentioned in Chapter 3, that the proposed model was deliberately trained for only two epochs. This choice was strategically made based on the observation that the F1 scores yielded their optimal performance within this limited training timeframe.

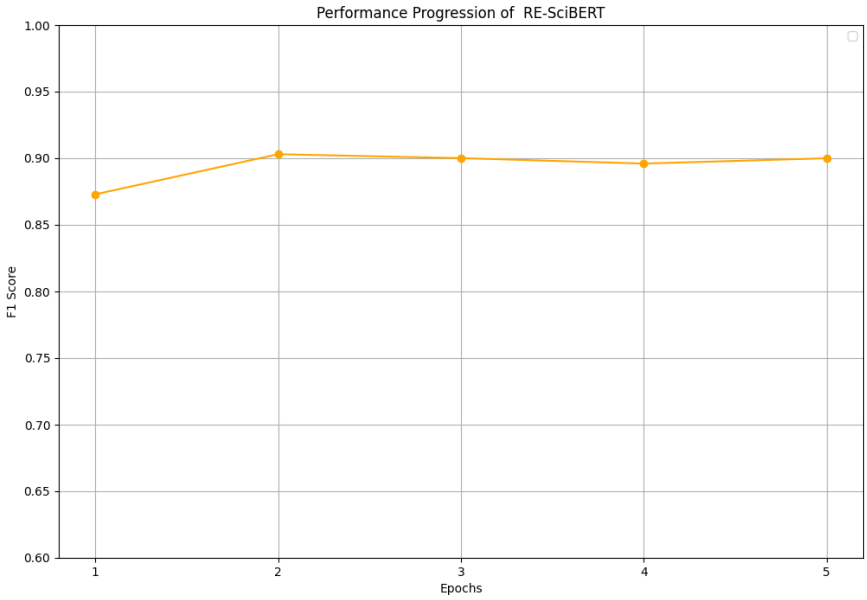


FIGURE 4.4 – RE-SciBERT F1 Score curve.

Table 4.3 presents a comprehensive breakdown of the fine-tuned RE-SciBERT model's performance on the test set. The classification report, generated, reveals valuable insights. Class CPR :4 exhibits the highest performance and holds the largest proportion among all relations. Conversely, CPR :3 stands out as the most challenging relation type for classification.

Label	**Precision**	**Recall**	**F1-score**
CPR :5	0.84	0.64	0.73
CPR :6	0.84	0.73	0.78
CPR :9	0.74	0.47	0.58
CPR :4	0.82	0.76	0.79
CPR :3	0.74	0.61	0.67

TABLE 4.3 – Performance breakdown of our proposed fine-tuned SciBERT model.

Figure 4.5 delves deeper into the prevalence of errors in CPR relation extraction by our fine-tuned SciBERT model on the test set. This is visualized using a confusion matrix,

where the predicted labels by the model are represented on the x-axis, and the true labels are on the y-axis. Each cell showcases the total number of relation instances for that specific combination of predicted and true labels. The color intensity within each cell is normalized by row, with darker shades of blue indicating a higher number of instances and lighter shades indicating fewer instances.

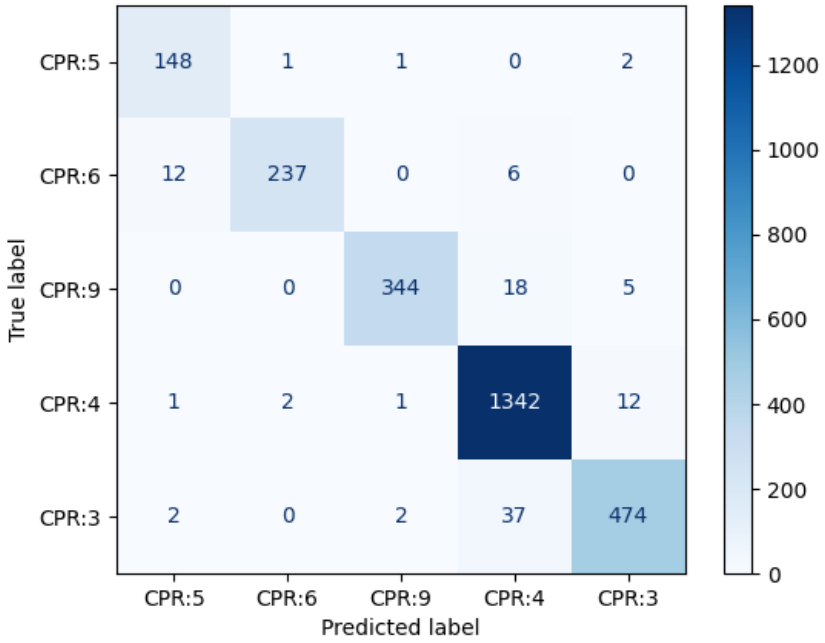


FIGURE 4.5 – Confusion matrix of the CPR.

4.4 Test of the proposed RE model

To evaluate the performance of our proposed RE model in the biomedical domain, we prepared a test set comprised of sentences containing placeholders for chemical and gene entities.

Table 4.4 presents the relationships extracted by our proposed RE-SciBERT model for the corresponding sentences. The results demonstrate a remarkable achievement, with the model successfully extracting all the intended relationships from the provided data.

Sentence	Real Relation	Our Model's Relation
We conclude that alprenolol and @CHEMICAL$ are competitive slowly reversible @GENE$ antagonists on rat left atria.	CPR :6	CPR :6
Eosinophils were isolated from peripheral blood, treated with either buffer or 10(-)10 M to 10(-)6 M FP in the presence of 10 pg/ml human recombinant interleukin-5 (@CHEMICAL$) and activated with @GENE$ (FMLP) + cytochalasin B (CB).	CPR :3	CPR :3
The nonselective and irreversible @CHEMICAL$ inhibitors, phenelzine (3-10 mg/kg), @GENE$ (1-3 mg/kg), and nialamide (30 mg/kg), decreased rates of responding maintained by ethanol reinforcement.	CPR :4	CPR :4
Parenteral administration of selective agonists of the delta-opioid receptor (SB 227122), mu-opioid receptor (codeine and hydrocodone), and @CHEMICAL$ (@GENE$) produced dose-related inhibition of citric acid-induced cough with ED (50) values of 7.3, 5.2, 5.1, and 5.3 mg/kg, respectively.	CPR :5	CPR :5

TABLE 4.4 – Testing the proposed Biomedical RE Model.

4.5 Conclusion

This chapter discussed the findings and analysis of a relation extraction model designed for ChemProt. We started by explaining how we set up the experiments.

Next, a detailed breakdown of the model's performance was provided in the results and discussion section.

We also included examples of how the model performed on test data. Overall, the fine-tuned SciBERT model proves to be an effective tool for extracting relationships from biomedical literature, with significant potential to aid in the advancement of biomedical research.

Conclusion and Future Perspectives

This dissertation introduced a model for relation extraction tasks within the biomedical domain, specifically targeting the extraction of chemical-protein interactions. The approach leverages the capabilities of transformer models, particularly SciBERT, enhanced by advanced deep learning techniques to achieve accurate identification and classification of these interactions.

Relation extraction is a crucial task in natural language processing that involves identifying and categorizing relationships between entities within a text. In the biomedical field, RE is instrumental for uncovering complex interactions, such as those between chemicals and proteins, which are vital for understanding biological processes and disease mechanisms.

The core of this research lies in the implementation of a transformer-based model, SciBERT, which has been fine-tuned to optimize its performance for the specific task of extracting chemical-protein interactions from biomedical literature. The model's construction involved the integration of language models, thorough preprocessing techniques, and precise fine-tuning procedures. Fine-tuning involves adjusting the pre-trained model on a specialized dataset to improve its accuracy and relevance for specific tasks. The dataset used in this study was meticulously curated to include a wide array of chemical-protein interactions, providing a robust foundation for training and evaluating the model.

The empirical results demonstrated the model's effectiveness, showcasing its promising potential in accurately extracting chemical-protein relationships from biomedical literature.

Future endeavors should focus on expanding the RE model's applicability beyond chemical-protein interactions. This necessitates training on diverse datasets encompassing

a wider range of biological and clinical relationships. Rigorous evaluation across various biomedical subfields will be crucial to ensure the model's versatility and generalizability. Additionally, exploration of transformer variants, such as ERNIE and BioBERT, holds immense potential. Investigating hybrid approaches that combine these models' strengths could lead to even more robust and accurate relation extraction capabilities. Systematic assessment of these models' efficacy in specific tasks will be paramount in identifying the most effective solutions for diverse biomedical applications.

Bibliography

[1] Mohammed Saber, Abdessamad El Rharras, Rachid Saadane, Hatim Kharraz Aroussi, and Mohammed Wahbi. Artificial neural networks, support vector machine and energy detection for spectrum sensing based on real signals. *International Journal of Communication Networks and Information Security (IJCNIS)*, 11(1) :52–53, April 2019.

[2] Khadidja METTAS and Maissa Fatna RAI. *A Deep Learning Model for Text-based CAPTCHA Breaking.* PhD thesis, University of Ghardaia, 2020.

[3] Chris Olah. Understanding lstm networks, 2015. https://colah.github.io/posts/2015-08-Understanding-LSTMs/.

[4] Liwei Hu, Jun Zhang, Yu Xiang, and Wenyong Wang. Neural networks-based aerodynamic data modeling : A comprehensive review. *IEEE Access*, PP :1–1, 05 2020.

[5] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, volume 30, 2017.

[6] Badiuzzaman Shuvo, Rifat Ahommed, Sakib Reza, and M.M.A. Hashem. Cnl-unet : A novel lightweight deep learning architecture for multimodal biomedical image segmentation with false output suppression. *Biomedical Signal Processing and Control*, 70, 07 2021.

[7] Martin Krallinger, Obdulia Rabal, Saber A. Akhondi, et al. Overview of the biocreative vi chemical-protein interaction track. In *Proceedings of the sixth BioCreative challenge evaluation workshop*, pages 141–146, 2017.

[8] Julien Delaunay, Hanh Thi Hong Tran, Carlos-Emiliano González-Gallardo, Georgeta Bordea, Nicolas Sidere, and Antoine Doucet. A comprehensive survey of document-

level relation extraction (2016-2023). *ACM Transactions on Information Systems*, 1(1) :35, October 2023. https ://doi.org/10.1145.

[9] Hadjer Khaldi. *Business relation extraction from texts*. PhD thesis, Université Paul Sabatier - Toulouse III, 2022. NNT : 2022TOU30325, tel-04186286f.

[10] Xu Wang, Kehai Chen, Lili Mou, and Tiejun Zhao. Document-level relation extraction with sentences importance estimation and focusing. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*, pages 2920–2929. Association for Computational Linguistics, 2022.

[11] Michael Banko, Michael J. Cafarella, Stephen Soderland, Matt Broadhead, and Oren Etzioni. Open information extraction from the web. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2670–2676, 2007.

[12] Anthony Fader, Stephen Soderland, and Oren Etzioni. Identifying relations for open information extraction. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1535–1545, 2011.

[13] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. Biobert : a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4) :1234–1240, 2020.

[14] Robert Leaman, Chih-Hsuan Wei, Aurélie Névéol, and Zhiyong Lu. Dnorm : Disease name normalization with pairwise learning to rank. *Bioinformatics*, 31(22) :3659–3667, 2015.

[15] Erik Cambria, Soujanya Poria, and Rajiv Bajpai. Senticnet 3 : A common and common-sense knowledge base for cognition-driven sentiment analysis. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2014.

[16] Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. The stanford corenlp natural language processing toolkit. In *Proceedings of the ACL (System Demonstrations)*, pages 55–60, 2014.

[17] Zhilin Yang, Diyi Yang, Xiaodong Yang, and Hui Yu. Finbert : A pre-trained language model for financial communications. In *Proceedings of the ACL*, pages 1302–1313, 2020.

[18] Alexander Yates, Michael Banko, Matthew J. Cafarella, Oren Etzioni, and Matt Broadhead. Real-time open domain question answering with dense-sparse phrase indexing. 2018.

[19] Mike Mintz, Steven Bills, Danah Boyd, M. A. Diwan, Anil Jain, Bhushan Thakkar, and Richard M. Wartell. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 1003–1011, 2009.

[20] Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao. Relation classification via convolutional deep neural network. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics : Technical Papers*, pages 2335–2344, 2014.

[21] Makoto Miwa, Mohit Bansal, and Yoshimasa Tsuruoka. End-to-end relation extraction using lstms on sequences and tree structures. *arXiv preprint arXiv :1601.00770*, 2016.

[22] Karoline B. Kuchenbaecker, John L. Hopper, Douglas F. Easton, Paul D. Pharoah, Alison M. Dunning, Manjeet K. Bolla, Qin Wang, Joe Dennis, Kyriaki Michailidou, Jonathan P. Tyrer, and et al. Risks of breast, ovarian, and contralateral breast cancer for brca1 and brca2 mutation carriers. *JAMA*, 317(23) :2402–2416, 2017.

[23] Guangyu Wu, Luonan Nie, Wen Zhang, and Jie Liu. Predicting protein–protein interactions from protein sequences using meta predictor. *Amino Acids*, 48(6) :1449–1461, 2016.

[24] John R. Vane, Ronald M. Botting, and John H. Botting. Aspirin and other anti-inflammatory drugs. In *Handbook of Experimental Pharmacology*, volume 33, pages 1–36. 1971.

[25] John R. Horn, Philip D. Hansten, and Lily N. Chan. Proposal for a new tool to evaluate drug interaction cases. *Annals of Pharmacotherapy*, 41(4) :674–680, 2007.

[26] P. Zhou, X.-L. Yang, X.-G. Wang, B. Hu, L. Zhang, W. Zhang, H.-R. Si, Y. Zhu, B. Li, C.-L. Huang, H.-D. Chen, J. Chen, Y. Luo, H. Guo, R.-D. Jiang, M.-Q. Liu, Y. Chen, X.-R. Shen, X. Wang, X.-S. Zheng, K. Zhao, Q.-J. Chen, F. Deng, L.-L. Liu, B. Yan, F.-X. Zhan, Y.-Y. Wang, G.-F. Xiao, and Z.-L. Shi. A pneumonia outbreak

associated with a new coronavirus of probable bat origin. *Nature*, 579(7798) :270–273, 2020.

[27] Iz Beltagy, Kyle Lo, and Arman Cohan. Scibert : A pretrained language model for scientific text. In *Proceedings of the EMNLP-IJCNLP*, pages 3615–3620, 2019.

[28] Jin-Dong Kim, Tomoko Ohta, Yoshimasa Tsuruoka, Yuka Tateisi, and Nigel Collier. Introduction to the bio-entity recognition task at jnlpba. In *Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing*, pages 70–75, 2011.

[29] Sampo Pyysalo, Tomoko Ohta, Ruchir Rak, Dan Sullivan, Chun-Nan Hsu, Filip Ginter, Tadayoshi Hara, Yi-Chen Hsiao, Yuji Matsumoto, Takeshi Miyao, and Nigel Collier. Overview of the bionlp shared task 2011. *BMC Bioinformatics*, 13(Suppl 11) :S1, 2012.

[30] Guoqian Jiang, Harold R. Solbrig, and Christopher G. Chute. Adept, a semantically-enriched pipeline for extracting adverse drug events from free-text electronic health records. *PloS One*, 11(5) :e0153018, 2016.

[31] Mohammad Khalilia, Sourav Chakraborty, Mihai Popescu, and Samaher A. Hasan. Predicting disease risks from highly imbalanced data using random forest. *BMC Medical Informatics and Decision Making*, 16(1) :1–15, 2016.

[32] A. Nikitin, S. Egorov, N. Daraselia, and I. Mazo. Pathway studio—the analysis and navigation of molecular networks. *Bioinformatics*, 19(16) :2155–2157, 2003.

[33] A. Krämer, J. Green, J. Pollard Jr, and S. Tugendreich. Causal analysis approaches in ingenuity pathway analysis. *Bioinformatics*, 30(4) :523–530, 2014.

[34] L. Luo, P.-T. Lai, C.-H. Wei, C.N. Arighi, and Z. Lu. Biored : A rich biomedical relation extraction dataset. *Briefings in Bioinformatics*, 2022.

[35] Sampo Pyysalo, Filip Ginter, Juho Heimonen, Jari Bjorne, Jorma Boberg, Jorma Jarvinen, and Tapio Salakoski. Bioinfer : a corpus for information extraction in the biomedical domain. *BMC Bioinformatics*, 8(1) :1–24, 2007.

[36] J. Li, Y. Sun, R.J. Johnson, et al. Biocreative v cdr task corpus : a resource for chemical disease relation extraction. *Database*, 2016 :baw068, 2016.

[37] W. S. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, 5(4) :115–133, 1943.

[38] Frank Rosenblatt. The perceptron : A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6) :386, 1958.

[39] Ms.Sonali. B. Maind. Department of information technology datta meghe institute of engineering, technology research, sawangi (m), wardha.

[40] Keiron O'Shea and Ryan Nash. An introduction to convolutional neural networks. 2015.

[41] Anthony Brabazon, Michael O'Neill, and Seán McGarraghy. *Natural Computing Algorithms*, volume 554. Springer, 2015.

[42] Jeffrey L. Elman. Finding structure in time. *Cognitive Science*, 14(2) :179–211, 1990.

[43] Sepp Hochreiter and Jürgen Schmidhuber. Lstm can solve hard long time lag problems. In *Advances in Neural Information Processing Systems*, volume 9, 1996.

[44] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv :1301.3781*, 2013.

[45] Jeffrey Pennington, Richard Socher, and Christopher 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.

[46] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. 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 2227–2237, 2018.

[47] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training, 2018.

[48] 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, pages 4171–4186, 2018.

[49] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, ..., and Veselin Stoyanov. Roberta : A robustly optimized bert pretraining approach. arXiv preprint arXiv :1907.11692, 2019.

[50] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. Xlnet : Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5753–5763, 2019.

[51] Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Hao Tian, Hua Wu, and Haifeng Wang. ERNIE : Enhanced representation through knowledge integration. arXiv preprint arXiv :1904.09223, 2019.

[52] Sultan Alrowili and K. Vijay-Shanker. Biom-transformers : Building large biomedical language models with bert, albert and electra. In *Proceedings of the 20th Workshop on Biomedical Language Processing*, pages 221–227, 2021.

[53] Hoo-Chang Shin, Yang Zhang, Evelina Bakhturina, Raul Puri, Mostofa Patwary, Mohammad Shoeybi, and Raghav Mani. Biomegatron : Larger biomedical domain language model. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4700–4706, 2020.

[54] Michihiro Yasunaga, Jure Leskovec, and Percy Liang. Linkbert : Pretraining language models with document links. *arXiv preprint arXiv :2203.15827*, 2022.