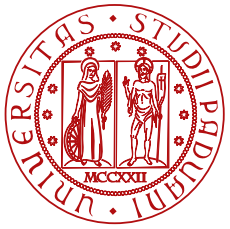

Università degli Studi di Padova

Dipartimento di Ingegneria dell'Informazione

Master Thesis in Computer Engineering

An Experimental Study on Bidirectional Encoder Representations from Transformers (BERT) for Named Entity Recognition and Relation Extraction

Master Candidate

Odai Mohammad

Supervisor

Giorgio Maria Di Nunzio

Date : 10 July

Academic Year

2023/2024

For the University of Padova that took me in and has been an incredible source of inspiration. A huge thanks to my amazing professor, Giorgio Maria Di Nunzio, whose guidance and support have been priceless. To my mom and my brother, your love and encouragement have meant the world to me, and I couldn't have done this without you. And to my awesome friends, thank you for the laughter, support, and for being there through all the ups and downs. You all rock! Of course, a special shoutout to my cat King,

Abstract

The rapid growth of digital content across platforms such as social media, news articles, academic publications, and online forums has resulted in an overwhelming volume of unstructured textual data. Extracting meaningful information from this data is critical for numerous applications, including information retrieval, knowledge base population, and automated question-answering systems. Named Entity Recognition (NER) and Relation Extraction (RE) are essential components in this process, enabling the identification of entities and the relationships between them. However, traditional models often fall short in handling the complexities of language, particularly domain-specific terminologies and intricate relational structures.

This thesis explores the application of Bidirectional Encoder Representations from Transformers (BERT), a state-of-the-art pre-trained language model, for NER and RE tasks. The primary objectives are to evaluate the performance of BERT-based models on domain-specific datasets, compare them with existing state-of-the-art techniques, and develop a framework for efficient training and application of these models across various contexts.

Our study involves a comprehensive experimental setup using diverse datasets, including scientific texts, to assess BERT's ability to handle specialized vocabularies and complex relational data. The methodology includes fine-tuning BERT models for NER and RE, implementing rigorous evaluation metrics, and comparing results with other contemporary models. We focus on reproducibility and robustness, ensuring that our findings are applicable across different domains and data types.

The findings reveal that while our BERT-based model may not always exceed the performance of current state-of-the-art models, it performs on par with them. Significantly, it achieves this with a more straightforward design and substantially lower computational overhead. This efficiency makes it an attractive option for practical scenarios where minimizing resource use and operational costs is crucial.

Contents

List of Figures

List of Tables

List of Code Snippets

List of Acronyms

IE Information Extraction

NER Named Entity Recognition

RE Relation Extraction

NLP Natural Language Processing

ACE The Automatic Content Extraction

FFN Feedforward Netwrok

KB Knowledge Base

LDC Linguistic Data Consortium

EDT Entity Detection and Tracking

LNK Entity Linking

RDC Relation Detection and Characterization

EDC Event Detection and Characterization

GPEs Geo-Political Entities

AI Artificial Intelligence

NYT dataset New York Times Relation Extraction Dataset

TACRED Text Analysis Conference Relation Extraction Dataset

TAC Text Analysis Conference

KBP Knowledge Base Population

NIST National Institute of Standards and Technology

LSTM Long Short-Term Memory

GCN Graph Convolutional Network

MRC Machine Reading Comprehension

DYGIE Dynamic Graph Interaction Extraction

BERT Bidirectional Encoder Representations from Transformers

AT Adversarial Training

PURE the Princeton University Relation Extraction system

BERT Bidirectional Encoder Representations from Transformers

ReLU Rectified Linear Unit

LLM Large Language Model

1 Introduction

In today's connected world, the explosion of textual data from platforms such as social media, news websites, academic journals, and online forums has resulted in a vast amount of unstructured or semi-structured information. Extracting meaningful insights from this vast corpus of data poses a significant challenge, which has led to the development of advanced Natural Language Processing (NLP) techniques. Among these, Information Extraction (IE) stands out as a pivotal tool for transforming raw text into structured, actionable knowledge, thus facilitating tasks ranging from information retrieval to automated knowledge base construction.

NER and RE are crucial components of IE, playing a vital role in identifying entities within text and determining the relationships between them. These tasks are fundamental in various applications, including knowledge base population, information retrieval, and question-answering systems. However, traditional IE models often struggle with the complexity of language, including domain-specific terminologies and the inherent variability and ambiguity of natural language, making them less effective in specialized or dynamic contexts.

The advent of deep learning, particularly the use of pre-trained language models such as Bidirectional Encoder Representations from Transformers (BERT) [1], has revolutionized the field of NLP. BERT's ability to understand context through bidirectional encoding makes it particularly well-suited for tasks like NER and RE, offering significant improvements over previous models that relied on simpler contextual representations. BERT's pre-training on large corpora allows it to capture intricate patterns and nuances in language, which are essential

for accurate entity recognition and relation extraction. This has led to substantial advancements in the automation of information extraction, enhancing the capability to process and understand complex textual data.

This thesis presents an experimental study on the application of BERT for NER and RE, with a specific focus on the reproducibility of the results and thorough evaluation across multiple datasets. The motivation behind this research is to explore the effectiveness of BERT in handling domain-specific language and complex relational structures found in scientific texts. By leveraging BERT's capabilities, we aim to enhance the extraction of entities and relations, which is critical for advancing knowledge in specialized fields such as biomedical research, legal document analysis, and technical literature.

The objectives of this study are threefold: First, to evaluate the performance of BERT-based models on NER and RE tasks using domain-specific datasets. Second, to compare the results of BERT-based models with other state-of-the-art techniques, assessing their robustness and generalizability. Third, to provide a framework for training efficient models tailored to various types of NER and RE tasks, thereby offering a scalable solution for future applications.

Through this research, we aim to contribute to the ongoing development of advanced IE methods that can effectively transform unstructured text into structured knowledge, thereby facilitating better understanding and utilization of textual data across different fields. The findings of this study are expected to provide valuable insights into the practical applications of BERT for NER and RE and highlight the potential for future innovations in NLP.

The rest of the thesis is organized as follows. In chapter 2 we review the existing literature on NER and RE, focusing on the challenges these tasks face and the various approaches developed to address them. It provides an overview of different datasets and models used in previous studies, highlighting their strengths and limitations. Chapter 3 presents the problem definition and the methodological framework employed in the study. It describes the design of the NER and RE models based on BERT, detailing the training processes, evaluation metrics, and the specific datasets used for experiments. Furthermore, in chapter 4 we details the experimental setup, including the datasets, evaluation metrics, and implementation details. It outlines the steps taken to train and test the models, discussing the configuration and parameters used in the experiments. Chapter 5 presents the results of the experiments, providing a comprehensive analysis of the model performance on different datasets. It compares the BERT-

based models with other state-of-the-art techniques, discussing the findings and their implications. Finally, in chapter 6, we summarize the key findings of the study, discussing the significance of the results and their contributions to the field. It outlines potential directions for future research, suggesting ways to further enhance the capabilities of BERT for NER and RE tasks.

2

Related Work

With the rise of the Internet, there has been a notable surge in digital text creation across various platforms such as social media, emails, blogs, news articles, publications, and online forums. This vast corpus of unstructured or semi-structured text harbors a wealth of information. IE is a pivotal tool in discerning and organizing meaningful insights from these textual sources, transforming them into structured data.

One way to represent information in text is in the form of entities and relations representing links between entities. Therefore, NER and RE emerge as particularly valuable techniques and key components of IE. They enable extracting pertinent entities and relationships within the text, facilitating the conversion of raw data into structured repositories of valuable information.

The NER task identifies entities from the text, and the RE task can identify relationships between those entities. Furthermore, end-to-end relation extraction aims to identify named entities and extract relations between them in one go. They are effectively modeling these two subtasks jointly [2], either by casting them in one structured prediction framework or performing multi-task learning through shared representations.

Many NLP applications can benefit from relational information derived from natural language [3], including Structured Search, Knowledge Base (KB) population, Information Retrieval, Question-Answering, Language Understanding, Ontology Learning, etc. Therefore these tasks have been studied extensively and many datasets have been created and many models have been proposed to tackle them.

2.1 NER and RE challenges

The NER and RE tasks face many challenges that need to be overcome. These challenges include and are not limited to:

- Domain-Specific Terminology and Context [4]: Adapting models to effectively handle domain-specific terminology, especially in specialized fields requires significant tuning and domain knowledge.
- Variability and Ambiguity in Text [5]: The inherent variability and ambiguity in natural language make it challenging to accurately identify and classify entities and relations, particularly in cases of sparse or implicit information.
- Data Scarcity and Annotation Quality: High-quality, annotated datasets are crucial for training effective models. However, the scarcity of such datasets in specific domains and the variability in annotation quality can hinder model performance and generalization.
- Cross-Domain and Cross-Linguistic Applicability: Developing models that perform well not only across different domains but also across languages is a significant challenge [6], requiring robust and adaptable methodologies.
- Integration of Knowledge Bases and External Information: Effectively integrating external knowledge bases and contextual information to improve the accuracy of NER and RE tasks remains a complex challenge [7].

2.2 Datasets

The exploration and understanding of complex textual data have significantly advanced with the development of NER and RE technologies. Central to these advancements are the diverse datasets that have been meticulously curated to train and evaluate these information extraction systems.

2.2.1 The Automatic Content Extraction dataset

The Automatic Content Extraction (ACE) Program was launched to boost the creation of technologies for processing language data automatically [8]. This included tasks like classifying, filtering, and choosing data based on its content and the meanings conveyed. The main aim of the ACE Program was to improve

technologies that could automatically recognize and describe these meanings, helping to enhance how machines understand natural language.

Central to the ACE Program were its research objectives: the detection and characterization of Entities, Relations, and Events. These objectives were meticulously addressed through the development of annotation guidelines, corpora, and other linguistic resources by the Linguistic Data Consortium (LDC), some in cooperation with the TIDES Program for supporting TIDES Extraction evaluations. The datasets produced under ACE, encompassing broadcast transcripts, newswire, and newspaper data in English, Chinese, and Arabic, became pivotal resources for training and testing in common research task evaluations.

The primary ACE annotation tasks were Entity Detection and Tracking (EDT), Relation Detection and Characterization (RDC), Event Detection and Characterization (EDC), and Entity Linking (LNK). EDT laid the groundwork by identifying entities within a document across mentions—named, nominal, or pronominal. Entities were classified into seven types—Person, Organization, Location, Facility, Weapon, Vehicle, and Geo-Political Entities (GPEs), with further distinctions into subtypes. This detailed entity annotation schema provided a robust foundation for subsequent tasks, enabling a nuanced understanding of text data.

The RDC task, introduced in the program's second phase, was pivotal in identifying and characterizing the relations between entities. This addition significantly expanded the scope of the ACE dataset, incorporating a variety of relations such as physical, social/personal, employment/membership, and more. The emphasis on capturing relations supported by textual evidence versus those inferred contextually introduced a layer of complexity, pushing forward the capabilities in relation extraction technologies.

With the introduction of EDC in ACE Phase 3, the program took on the new challenge of identifying and categorizing events in which entities participate. This expanded the dataset's utility by providing insights into interactions, movements, transfers, creations, and destructions depicted in text, along with event arguments and attributes based on type-specific templates. Later phases further enriched the dataset with additional event types and characterized relations between events, offering an even more comprehensive resource for NER and RE tasks.

The significance of the ACE dataset to NER and RE tasks lies in its comprehensive coverage of entities, relations, and events, making it a cornerstone in the

development of technologies for information extraction. By providing a structured framework for annotating and understanding complex language data, the ACE dataset has been instrumental in advancing research and applications in NER and RE, enabling more sophisticated and nuanced language processing capabilities.

2.2.2 The SciERC dataset

The SciERC dataset, meticulously crafted from the domain of scientific research papers, particularly in the field of Artificial Intelligence (AI), represents a significant advancement in the realm of NER and RE tasks [9]. Developed by the Allen Institute for AI, SciERC's primary objective is to facilitate the extraction of scientific entities, their relationships, and events from AI research literature, thereby enabling a deeper understanding and structuring of scientific knowledge. This dataset emerges from the recognition of the unique challenges presented by scientific texts, which include domain-specific terminology, complex entity relations, and the nuanced depiction of scientific events and processes.

SciERC is distinguished by its focus on scientific texts, comprising 500 abstracts from AI conference proceedings, annotated for entities, relations, and coreference clusters. Entities within SciERC are categorized into specific types such as tasks, methods, metrics, materials, and others relevant to scientific discourse. This categorization facilitates a granular understanding of the scientific narrative, allowing for the extraction of nuanced information regarding the methodologies, tools, and outcomes prevalent within AI research. Furthermore, the dataset annotates relations between these entities, providing insights into the interdependencies and interactions that define scientific innovation. Such detailed annotation makes SciERC an invaluable resource for developing NER and RE models tailored to the scientific domain.

The significance of the SciERC dataset to NER and RE tasks extends beyond its domain-specific focus. By offering a structured framework for analyzing scientific texts, SciERC enables the development of models capable of navigating the complexities inherent in scientific literature. These models are not only instrumental in extracting information from research papers but also in facilitating the synthesis of scientific knowledge, contributing to meta-analyses, systematic reviews, and the construction of scientific knowledge graphs. In this way, SciERC supports the broader objective of making scientific knowledge more accessible

and understandable, both to machines and to humans.

The connection between the SciERC dataset and our task is particularly poignant. Given the thesis's focus on extracting structured information from research papers within a specific scientific domain, SciERC provides a relevant model for addressing similar challenges in our research. The methodologies and insights gained from working with the SciERC dataset can inform the development of specialized NER and RE models for fields other than AI, enabling the extraction of entities and relations specific to those fields. Moreover, the success of models trained on SciERC underscores the potential for applying advanced NER and RE techniques to a wide range of scientific disciplines, thereby enhancing the accessibility and interoperability of scientific knowledge across domains.

The SciERC dataset represents a pivotal resource for advancing NER and RE tasks within the scientific domain. Its focus on AI research literature not only addresses the specific challenges of scientific text analysis but also offers a blueprint for extending these capabilities to other scientific disciplines. By enabling the development of models that can accurately identify and relate entities within scientific texts, SciERC contributes to the broader goal of structuring scientific knowledge, making it more navigable and comprehensible for both academic and practical purposes.

2.2.3 Other datasets

In our research, we have come to study other datasets that must be mentioned. Those datasets were used as a sanity check for our work. We relied on those datasets to prove that our models could be generalized and used for NER and RE tasks in other domains.

The New York Times Relation Extraction Dataset

The New York Times Relation Extraction Dataset (NYT dataset) is a prominent resource for RE, offering a comprehensive collection of news articles for the development and testing of RE models. Originating from a collaboration between the New York Times and Google, this dataset encompasses a vast array of articles published by the New York Times, annotated with both entities and the relations between them [10]. The primary objective of the NYT dataset is to support the extraction of semantic relationships within text, facilitating a deeper

understanding of the interconnectedness of entities as reported in journalistic content.

The dataset is characterized by its diverse coverage of topics, including politics, sports, culture, and more, reflecting the wide-ranging nature of news reporting. This diversity presents unique challenges and opportunities for RE, requiring models to adapt to various contexts and entity types. Each article within the dataset is annotated with detailed information about entities and the specific relations that link them, providing a rich ground for training sophisticated RE models capable of identifying and classifying a wide range of relation types.

The significance of the NYT dataset to the RE task lies in its real-world applicability and the complexity of its textual content. Working with this dataset enables researchers to hone RE models on text that encapsulates a broad spectrum of human activity and knowledge, mirroring the complexity and nuance of natural language used in daily news cycles. Additionally, the NYT dataset serves as a benchmark for evaluating the performance of RE models, offering a standard against which to measure progress in the field.

The NYT dataset exemplifies the application of RE techniques to general-domain text. The exploration of this dataset highlights the adaptability of RE methodologies across different textual domains, underscoring the potential for leveraging insights gained from working with the NYT dataset to enhance RE approaches tailored to scientific literature. This cross-domain exploration illustrates the broad applicability of RE technologies and the importance of diverse datasets in advancing the field.

Text Analysis Conference Relation Extraction Dataset (TACRED)

The Text Analysis Conference (TAC) Relation Extraction Dataset is a crucial dataset in the domain of Relation Extraction, developed under the auspices of the TAC Knowledge Base Population (KBP) evaluations. Managed by the National National Institute of Standards and Technology (NIST), the TAC KBP evaluations are designed to foster research and development in the field of information extraction, with a focus on building comprehensive knowledge bases from unstructured text [11]. The TAC RE dataset specifically aims to advance the state of RE technology by providing a set of documents annotated with entities and their relations, serving as both a training and evaluation resource for RE

systems.

The dataset encompasses a diverse collection of texts sourced from newswire and web texts, including a wide range of topics and entity types. Entities within the dataset are meticulously annotated, and the dataset identifies various types of semantic relations that occur between these entities, such as affiliation, personal/social relationships, and organizational roles, among others. This rich annotation scheme allows for the detailed examination and modeling of complex relationships within natural language, making the TAC RE dataset an invaluable resource for researchers and developers working on advanced RE systems.

The significance of the TAC RE dataset extends beyond its comprehensive annotations; it also serves as a benchmark for evaluating the performance of RE systems in a competitive and collaborative environment. Through the annual TAC KBP evaluations, participating systems are assessed on their ability to accurately identify and characterize relations between entities, fostering innovation and progress in the field. The dataset not only facilitates the development of more sophisticated and accurate RE models but also promotes the exploration of new methodologies and approaches in knowledge base population.

Including this dataset in our research underscores its role in pushing the boundaries of RE technology. The challenges and solutions encountered in the TAC RE dataset provide a valuable perspective on the adaptation of RE techniques to domain-specific needs, demonstrating how RE technologies can be leveraged to extract structured information from diverse sources of text.

2.3 Exisitng NER and RE Models

Many models have been proposed for tackling NER and RE tasks. And in recent years there's been an emphasis on joint models. Join models are designed to perform join extraction of entities and relations [2] at the same time. We can group existing joint models into two categories: structured prediction and multi-task learning.

2.3.1 Structured prediction models

Structured prediction approaches cast the two tasks into one unified framework, although it can be formulated in various ways.
Li and Ji [12] proposed an action-based system that identifies new entities as

well as links to previous entities, Zhang et al. [13]; A novel and impactful methodology for the incremental joint extraction of entity mentions and relations. Their approach diverges from traditional methods by employing a structured perceptron with beam-search, moving away from token-based tagging to a segment-based decoder inspired by semi-Markov chains. This shift allows for the utilization of global features as soft constraints, effectively capturing the interdependencies between entities and relations. Their research, conducted on the ACE dataset, demonstrates significant advancements over existing pipelined approaches. By formulating the problem as one of structured prediction, their model adeptly captures the linguistic and logical nuances inherent in complex textual relationships, thereby addressing the limitations of sequential classification steps that fail to model long-distance and cross-task dependencies. The introduction of novel global features based on soft constraints over the entire output graph structure marks a significant contribution to the field, showcasing the potential for improved accuracy and efficiency in the extraction tasks. Li and Ji's work stands as a pivotal reference in the exploration of joint models and global features for enhancing entity and relation extraction, offering valuable insights and methodologies that could be adapted and extended within the context of our research.

Wang and Lu [14] adopt a table-filling approach as proposed in (Miwa and Sasaki [15]); This distinct approach departs from traditional single-encoder methods. Their method introduces two specialized encoders: a table encoder and a sequence encoder, designed to synergize in the representation learning process for NER and RE. This allows each encoder to focus on the unique aspects of its task—capturing task-specific information effectively—while benefiting from the interaction between the two to enhance overall performance. They leverage multi-dimensional recurrent neural networks to better utilize the structural information within the table representation, addressing a common limitation in existing methods that often overlook or underutilize such information. Furthermore, they exploit the pairwise self-attention weights from pre-trained models like BERT [16] to enrich their model's understanding of word-word interactions, a strategy not previously employed for table representations in this context. Their experiments across several standard datasets show significant improvements over existing approaches, particularly highlighting the advantage of dual encoders over traditional single-encoder frameworks. This work not

only sets new state-of-the-art performance benchmarks but also opens up new avenues for leveraging the inherent structure in linguistic data for information extraction tasks.

Katiyar and Cardie [17] and Zheng et al. [18] introduced an approach based on sequence-tagging for the joint extraction of entity mentions and relations, each contributing novel methodologies to the domain of information extraction. Katiyar and Cardie introduce an attention-based recurrent neural network model that leverages Long Short-Term Memory (LSTM) networks to extract semantic relations between entity mentions without relying on dependency trees. Their model is distinct for its direct addressing of the relation extraction task by integrating attention mechanisms with LSTMs, enabling the model to focus on relevant parts of the text to better identify relationships between entities, even when they are not adjacent. This approach sidesteps the need for dependency tree information, making it more broadly applicable, especially for languages or domains where dependency parsing might be less accurate or entirely unavailable. Their experiments on the ACE dataset demonstrate significant improvements over previously established feature-based joint models, highlighting the efficacy of their methodology in enhancing the accuracy of both entity and relation extraction tasks.

Zheng et al. propose a different take on sequence tagging by introducing a novel tagging scheme that converts the joint task of entity and relation extraction into a single tagging problem. This simplifies the traditionally complex process of first identifying entities and then classifying relations between them. Their end-to-end model, also based on LSTM networks, directly extracts entities and their relations without the need for separate entity recognition and relation classification stages. By treating the problem as a tagging issue, they manage to avoid the error propagation and complexity associated with pipelined and feature-based methods. Their approach not only demonstrates superior performance on a public dataset produced by distant supervision methods but also underscores the potential of tagging-based methods in streamlining the extraction process and improving result accuracy.

These contributions represent significant advancements in the field of information extraction, particularly in the context of NER and RE. They offer insights into the potential of neural network architectures and tagging schemes to simplify and enhance the joint extraction of entities and relations, paving the way

for more efficient and accurate extraction methodologies suitable for a wide range of applications.
Sun et al. [19] and Fu et al. [20] used a graph-based method to predict entity and relation types, offering significant advancements in joint entity and relation extraction tasks. Sun et al. introduced a novel Graph Convolutional Network (GCN) approach that operates on an entity-relation bipartite graph, designed to perform joint inference on entity types and relation types within a unified framework. This method significantly outperformed existing joint models in entity performance while maintaining competitive relation performance on the ACE05 dataset. The key to their approach was the introduction of a binary relation classification task that allowed for more efficient and interpretable use of the entity-relation bipartite graph structure.

Fu et al. presented GraphRel, an end-to-end relation extraction model employing GCNs to jointly learn named entities and relations. By considering both the interaction between named entities and relations and the implicit features among all word pairs in the text, GraphRel demonstrated substantial improvements in predicting overlapping relations compared to previous sequential approaches. Their graph-based strategy, which utilized both linear and dependency structures, alongside a complete word graph to extract features, resulted in high precision and a significant increase in recall, setting new state-of-the-art benchmarks for relation extraction on public datasets like NYT and WebNLG.

These contributions reflect a deeper understanding of how entities and relations interconnect within text, underscoring the potential of graph-based models to capture complex relationships more effectively than traditional methods. Through the integration of GCNs and strategic graph construction, both approaches highlight the evolving landscape of natural language processing, where the interconnectedness of textual elements is increasingly recognized and leveraged for more nuanced and accurate information extraction.

and, Li et al [21] project the task onto a multi-turn question answering problem, transforming the entity and relation extraction process into an innovative QA framework. This paradigm shift offers several advantages: it encodes specific class information for the desired entity or relation through the formulation of questions, naturally incorporates joint modeling of entities and relations, and leverages advanced Machine Reading Comprehension (MRC) models. Their approach not only significantly outperforms existing models on benchmark

datasets like ACE and CoNLL04 but also establishes new state-of-the-art results, highlighting its effectiveness in accurately identifying structured information from text. Moreover, Li et al. introduce a complex dataset, RESUME, requiring multi-step reasoning for entity dependency construction, further demonstrating the model's capability in handling intricate entity-relation mappings. This multi-turn QA framework marks a substantial advance in entity-relation extraction, showing promise for more nuanced and accurate information extraction from unstructured data.
All of these approaches need to tackle a global optimization problem and perform joint decoding at inference time, using beam search or reinforcement learning.

In general, structured prediction models are challenged by the complexity in modeling interdependencies. These models attempt to capture the complex interdependencies between entities and relations within a single framework, which can be computationally intensive and challenging to optimize. In addition, they also have to deal with the complexity of joint decoding. Performing joint decoding at inference time, such as using beam search or reinforcement learning, adds to the computational overhead and complexity.

2.3.2 Multi-task learning models

This family of models essentially builds two separate models for entity recognition and relation extraction and optimizes them together through parameter sharing. Miwa and Bansal [22] propose to use a sequence tagging model for entity prediction and a tree-based LSTM model for relation extraction. The two models share one LSTM layer for contextualized word representations, and they find sharing parameters improves performance (slightly) for both models. Their innovative approach captures both word sequence and dependency tree substructure information, integrating bidirectional tree-structured LSTM-RNNs on top of bidirectional sequential LSTM-RNNs. This allows for a single model to jointly represent entities and relations with shared parameters, improving over the state-of-the-art feature-based models on end-to-end relation extraction tasks.

The F1-score, a measure of a model's accuracy that considers both precision and recall, shows significant improvement with Miwa and Bansal's model. Specifically, their model demonstrates substantial error reductions in F1-score

on the ACE2005 and ACE2004 datasets, which are standard benchmarks for evaluating entity and relation extraction systems. The ACE2005 dataset, created as part of the Automatic Content Extraction (ACE) program, includes annotated texts for entities, relations, and events across a variety of domains such as newswire, broadcast news, and conversational telephone speech. Similarly, the ACE2004 dataset provides annotated examples for entities and their relationships, serving as an essential resource for training and evaluating models in natural language understanding tasks. Both datasets are crucial for advancing research in entity recognition and relation extraction, providing diverse and challenging examples for comprehensive model evaluation.

The approach of Bekoulis et al. [23] is similar except that they model relation classification as a multi-label head selection problem. Note that these approaches still perform pipelined decoding: entities are first extracted and the relation model is applied on the predicted entities. In their work on adversarial training for multi-context joint entity and relation extraction, Bekoulis et al. extend a baseline joint model that tackles NER and RE simultaneously, by introducing Adversarial Training (AT) as a regularization method. This technique enhances the model's robustness by incorporating small perturbations in the training data, thereby improving the state-of-the-art effectiveness across several datasets and languages. Their model successfully addresses the complexities of extracting multiple relations per entity by modeling relation extraction in a multi-label setting, allowing for a more nuanced understanding of the text. Additionally, their innovative use of AT demonstrates a significant improvement in the joint extraction task's effectiveness, showcasing the potential of adversarial examples in NLP to refine and strengthen model performance.
Dynamic Graph Interaction Extraction (DYGIE) and DYGIE++ (Luan et al. [24]; Wadden et al. [25]), build on recent span-based models for coreference resolution (Lee et al. [26]) and semantic role labeling (He et al. [27]). The key idea of their approaches is to learn shared span representations between the two tasks and update span representations through dynamic graph propagation layers. DYGIE++ extends upon these concepts by incorporating event extraction into its multi-task framework, utilizing both local (within-sentence) and global (cross-sentence) context to enumerate, refine, and score text spans. The system dynamically constructs graphs of spans, with edges representing task-specific relations, allowing for efficient global context modeling. This is achieved by refining initial contextualized embeddings, such as those from BERT, with task-

specific message updates propagated across the span graph.

The DYGIE++ framework demonstrates its effectiveness by achieving state-of-the-art results across several information extraction tasks and datasets, showcasing its capability to handle complex interdependencies among entities, relations, and events. The integration of BERT encodings enables the model to capture significant contextual relationships, including those extending beyond single sentences. Additionally, dynamic span graph updates further enhance the model's ability to incorporate cross-sentence dependencies, which is particularly beneficial for tasks in specialized domains. For example, leveraging predicted coreference links through graph propagation can help disambiguate challenging entity mentions by providing additional contextual clues.

A comprehensive evaluation of the DYGIE++ framework across different datasets reveals that its general span-based approach produces significant improvements in entity recognition, relation extraction, and event extraction tasks. The framework benefits from both types of contextualization methods—BERT encodings for capturing immediate and adjacent-sentence context, and message passing updates for modeling long-range cross-sentence dependencies. These findings underscore the importance of effectively integrating both local and global contextual information in a unified architecture to enhance performance on a range of information extraction tasks, making DYGIE++ a powerful tool for advancing research in this area.

A more recent work Lin et al. [28] further extends DYGIE++ by incorporating global features based on cross-substask and cross-instance constraints. They propose a joint neural framework named ONEIE, which aims to extract the globally optimal Information Extraction (IE) result as a graph from an input sentence. This framework performs IE in four stages: encoding the given sentence as contextualized word representations; identifying entity mentions and event triggers as nodes; computing label scores for all nodes and their pairwise links using local classifiers; and finally, searching for the globally optimal graph with a beam decoder. At the decoding stage, they introduce global features to capture the intricate cross-subtask and cross-instance interactions. Their experimental results demonstrate that adding these global features significantly improves the performance of their model, achieving new state-of-the-art results on all subtasks. Unlike previous models that use separate local task-specific classifiers in their final layer without explicitly modeling the dependencies among tasks and instances, ONEIE extracts a unified graph representation of the input sentence,

effectively capturing and leveraging the interdependencies among different IE components. This advancement underscores the importance of considering the holistic context of information in IE tasks, marking a significant step forward in the development of more integrated and contextually aware IE systems.
Zhong et al. [2] introduced the Princeton University Relation Extraction system (PURE), a similar approach. However, it is much simpler and performs better. Their model challenges the longstanding belief in the superiority of complex joint models for entity and relation extraction tasks. Through their research, they illuminate the effectiveness of a straightforward pipelined approach that employs two independent encoders for entity recognition and relation extraction, both built upon deep pre-trained language models. Their method deviates from the common practice of intricate joint modeling, advocating instead for simplicity and directness in treating the tasks sequentially. This simplicity, coupled with meticulous analyses on standard benchmarks like ACE04, ACE05, and SciERC, not only sets new state-of-the-art performances with absolute improvements in relation to F1 scores but also demonstrates the critical importance of learning distinct contextual representations for entities and relations. Furthermore, their investigation into incorporating entity information early in the relation model underscores the potential of a more focused approach to enhancing performance.

Their work significantly contributes to the discourse on the efficiency of information extraction models, showing that a model's complexity does not necessarily equate to its effectiveness. By simplifying the process into two distinct phases and ensuring each phase is optimized for its specific task, they reveal an often-overlooked aspect of model design: the power of specialization and focused optimization. Their findings suggest that the interactions between entities and relations, previously believed to be best captured jointly, can be effectively understood through a well-structured sequential approach. This revelation opens new avenues for future research in information extraction, particularly in exploring how different tasks within this domain can be optimized individually for better overall performance.

Moreover, the authors explored the utility of pre-trained language models as a foundation for both encoders bringing to light the substantial impact of these models in extracting meaningful insights from text. By leveraging such powerful models, they manage to streamline the extraction process and ensure that their approach remains flexible and robust across various datasets. This

adaptability, combined with the method's simplicity, marks a significant step forward in information extraction research. It prompts a reevaluation of current methodologies and suggests that the field might benefit from a shift towards simpler, more focused models that capitalize on the advancements in language modeling and representation learning.

However impressive, their model still faces several challenges. These include the potential for reduced effectiveness on rare or unseen entities, increased computational demands due to its complexity, reliance on accurate entity type identification, and difficulties in handling ambiguous contexts or adapting to various languages and domains. Moreover, the model risks overfitting to training data and may present challenges in interpretability, making it harder to understand how it makes decisions. Addressing these issues is essential for optimizing the model's performance and applicability across diverse datasets and settings.

3 Methodology

In this chapter, we provide a formal definition of the problem of joint entity and relation extraction in section [3.1]. Then, in section [3.2], we describe our approach in detail. First explaining the entity model in section [3.2.1], and then describing the relation model in section [3.2.2]. Finally, we explain the training and inference processes in section [3.2.3].

3.1 Problem Definition

Given X an input sentence consisting of n tokens $x_1, x_2, ..., x_n$. Let $S = s_1, s_2, ..., s_m$ be all the possible spans in X of up to length L and $START(i)$ and $END(i)$ denote start and end indices of s_i. The problem can be decomposed into two sub-tasks:

Named entity recognition Let E denote a set of pre-defined entity types. The named entity recognition task is, for each span $s_i \in S$, to predict an entity type

$$y_e(s_i) \in E,$$

or, span s_i is not an entity:

$$y_e(s_i) \notin E$$

The output of the task is

$$Y_e = (s_i, e) : s_i \in S, e \in E$$

.

Relation extraction Let R denote a set of predefined relation types. The task is, for every pair of spans $si \in S$, $sj \in S$, to predict a relation type

$$y_r(s_i, s_j) \in R,$$

or, there is no relation between them:

$$y_r(s_i, s_j) \notin R$$

. The output of the task is

$$Y_r = (s_i, s_j, r) : s_i, s_j \in S, r \in R$$

3.2 Our Approach

We based our approach on the state of the art proposed by Zhong et al. [2]. The simplicity of their model and its performance make it a prime candidate for NER and RE on new datasets. Therefore we first reproduced their results on the SciERC dataset. Then, we prove that their model can be generalized to other datasets. The next chapter will dive deeper into our experiments and demonstrate how we trained and evaluated their model on new NER and RE datasets. Namely the NYT (See section 2.2.3.1) and TACRED(See section 2.2.3.2) datasets. The goal is to have a framework that, given any NER and RE dataset, can be easily used to train NER and RE models.

3.2.1 Entity model

The entity model is inspired by previous research (Lee et al. [26]; Luan et al. [24]; Wadden et al. [25]). It begins by using a pre-trained language model, such as BERT, to understand the context of each word in a sentence. For any given segment of text, known as a 'span', we create a context representation χ_t for each input token x_t. This is done by combining the context of the span's first and last words with additional features that capture the span's length. We then use this combined information to calculate the likelihood of each entity type represented by this span.

Formally, Let S be an input sentence, and s_i a span of S $s_i \in S$, we can define the span representation $h_e(s_i)$ as [2]:

$$h_e(s_i) = [x_{START(i)}; x_{END(i)}; \phi(s_i)],$$

where $\phi(s_i) \in R_F^d$ is a representation of the span width features. Finally, $h_e(s_i)$ becomes the input to a Feedforward Netwrok (FFN) [29] that will predict the entity type. Or, more precisely, its probability distribution:

$$e \in E \cup \epsilon : P_e(e|s_i)$$

.

3.2.2 Relation Model

The relation model function is to take two spans s_i, s_j, which are the 'subject' and 'object' as input and output a relation between them. Or output ϵ if there's no relation. Most works we examined (Luan et al. [24]; Wadden et al. [25]) use the same span representations $h_e(s_i)$, $h_e(s_j)$ in the relation model to predict the relation between s_i and s_j . Hover, Zhong et al. [2] suggest that while these representations can understand the context surrounding each entity independently, they might not effectively identify the connections or relationships between pairs of text segments. They also point out that using the same contextual information for different pairs of text segments might not always be the best approach. For example, the phrase "is a" is important for recognizing the relationship between MORPA and PARSER in Figure 3.1, but does not help in understanding the connection between MORPA and TEXT-TO-SPEECH.

Instead, Zhong et al. [2] propose a relation model that looks at each pair of spans separately and adds specific markers in the initial processing stage. These markers indicate which span is the subject and which is the object, as well as their types, to improve the model's understanding. Formally, Let X be an input sentence and s_i , s_j be a pair of subject-object spans and $ei, ej \in E \cup \epsilon$ are their types respectively. Then we define text markers as $\langle S : e_i \rangle$, $\langle /S : e_i \rangle$, $\langle O : e_j \rangle$, and $\langle /O : e_j \rangle$, and embedded them into X before and after s_i and s_j (Figure 1 (b)). Let $\widehat{X}$ be the new sequence after inserting the markers:

$$\widehat{X} = ...\langle S : e_i \rangle, x_{START(i)}, ..., x_{END(i)}, \langle /S : e_i \rangle ...\langle /O : e_j \rangle, x_{START(j)}, ..., x_{END(j)} \langle /O : e_j \rangle$$

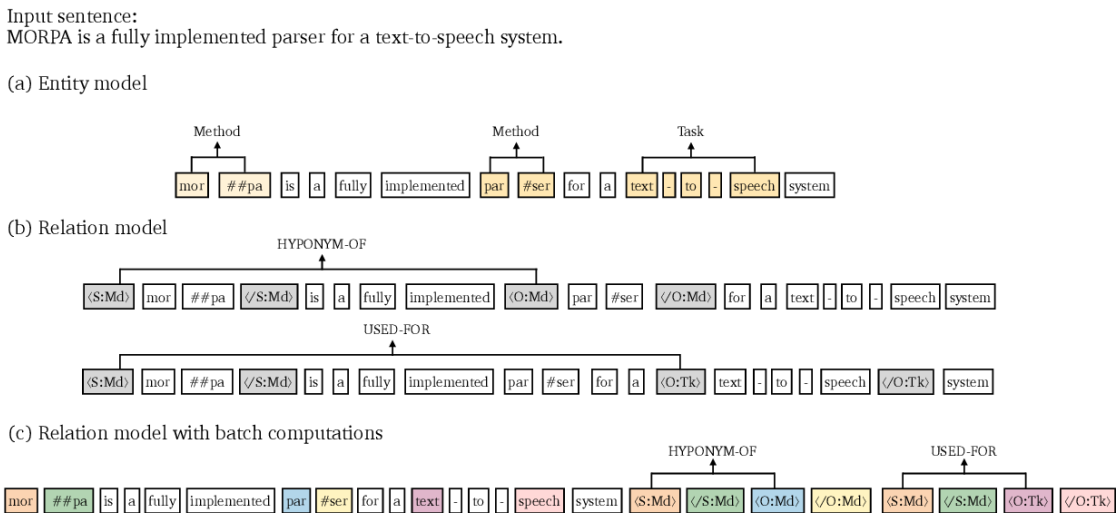

Figure 3.1: An input sentence from the SciERC dataset. Luan et al. [24]

Next, we use another pre-trained encoder on $\widehat{X}$ and we refer to its output representations with by $\widehat{x}_t$. We combine the outputs from their starting points to understand the relationship between the two spans. This gives us a combined representation:

$$h_r(s_i, s_j) = \left[\widehat{x}_{\widehat{START(i)}}; \widehat{x}_{\widehat{START(j)}}\right]$$

where $\widehat{START(i)}$ and $\widehat{START(j)}$ are the indices of $\langle S : e_i \rangle$ and $\langle O : e_j \rangle$ in $\widehat{X}$. Finally, $h_r(s_i, s_j)$ will be the input to an FFN that will predict the relation between s_i and s_j:

$$r \in R \cup \epsilon : P_r(r|s_i, s_j)$$

.

The approach of using special markers to identify subjects and objects in a text is not particularly novel and has been explored before in classification studies (Zhang et al. [30]; Soares et al. [31]; Peters et al. [32]). However, these studies usually focus on classifying the relationship between one pair of subjects and objects within a sentence Zhang et al. [33], such as in the TACRED dataset(See section 2.2.3.2). The effectiveness of this method has not been fully tested in the end-to-end setting like the Zhong et al. [2] hope to classify the relations amongst more entity mentions. They saw a significant improvement in their solution, strengthening the idea that different context-aware representations are invaluable for understanding the relations between entities pairs in one example.. Furthermore, Zhang et al. [30]; Soares et al. [31] used untyped only markers (e.g., $\langle S :\rangle$, $\langle /S :\rangle$) and previous end-to-end models (e.g., (Wadden et

al. [25])) only inject the entity type information into the relation model through auxiliary losses. Zhong et al. [2] found that injecting type information at the input layer is very helpful in distinguishing entity types — for example, whether "Disney" refers to a person or an organization— before trying to understand the relations.

3.2.3 Training and Inference

We adapt two pre-trained language models by fine-tuning them with task-specific loss functions, employing cross-entropy loss [2], for both the entity and relation extraction models. This equation penalizes the model more heavily when its predicted probability for the true entity type is lower, encouraging the model to correctly recognize entity types.

$$\mathcal{L}_e = -\sum_{s_i \in S} \log P_e(e_i^* | s_i)$$

Where $\mathcal{L}_e$ represents the cross-entropy loss function for the entity model. It is calculated by summing the negative log probabilities of the true (gold) entity types (e_i^*) for all spans (s_i) in the dataset (S). The probability $P_e(e_i^*|s_i)$ reflects how likely it is that the span s_i corresponds to its gold entity type e_i^* according to the model.

$$\mathcal{L}_r = -\sum_{s_i, s_j \in S, s_i \neq s_j} \log P_r(r_{i,j}^* | s_i, s_j)$$

Where $\mathcal{L}_r$ represents the cross-entropy loss function for the relation model. Similar to $\mathcal{L}_e$, this loss function sums the negative log probabilities that the model assigns to the true (gold) relation types ($r_{i,j}^*$) between pairs of spans (s_i, s_j) in the dataset (S), where $s_i \neq s_j$. The probability $P_r(r_{i,j}^*|s_i, s_j)$ indicates the model's confidence that the correct relation between the spans $s_i$and s_j is $r_{i,j}^*$.

where e_i^* represents the gold entity type of s_i and $r_{i,j}^*$ represents the gold relation type of span pair s_i, s_j in the training data. For training the relation model, we only consider the gold entities $S_G \subset S$ in the training set and use the gold entity labels as the input of the relation model. During inference, we first predict the entities by taking $y_e(s_i) = argmax_{e \in \varepsilon \cup \{\epsilon\}} P_e(e|s_i)$. Denote $S_{pred} = s_i : y_e(s_i) \neq \epsilon$, we enumerate all the spans $s_i, s_j \in S_{pred}$ and use $y_e(s_i), y_e(s_j)$ to construct the input for the relation model $P_r(r|s_i, s_j)$.

In training the relation model, we focus exclusively on the spans marked as entities according to the gold standard S_G, which is a subset of all spans S in the dataset ($S_G \subset S$). This allows the model to learn from the most relevant examples, using the gold entity labels to understand relationships within the text.

During the inference phase, the model predicts entities by determining the most likely entity type for each span s_i. That's done by taking

$$y_e(s_i) = argmax_{e \in \varepsilon \cup \{\epsilon\}} P_e(e|s_i),$$

where ε includes all possible entity types and ϵ indicates a non-entity. Spans not identified as entities are filtered out, creating a set $S_{pred} = s_i : y_e(s_i) \neq \epsilon$ of spans predicted to be entities.

With S_{pred} established, the model then examines every possible pair of spans within it to predict their interrelations. For each span pair s_i, s_j, it employs the predicted entity types as inputs to the relation model $P_r(r|s_i, s_j)$. This step utilizes the insights gained from entity prediction to enhance the accuracy of relation extraction, aiming to comprehensively map out the network of relationships among identified entities in the text. This process underscores the model's integrated approach, leveraging entity predictions to inform and refine relation extraction, thus creating a cohesive understanding of the textual content.

4

Experiments

This chapter details the datasets we used for the training and evaluation of our model in Section 4.1. In section 4.2 we explain the evaluation metrics used to evaluate our model. Finally, we explain the technical implementation used in our experiments in Section 4.3. The objective is to assess the robustness and adaptability of our model across different domains and setups, thereby demonstrating its practical utility in real-world applications.

4.1 Datasets

Our models were trained and evaluated using three primary datasets, each chosen for its unique characteristics and relevance to our research goals: the SciERC dataset (See section 2.2.2), the NYT relation extraction dataset (See section 2.2.3.1), and the TACRED dataset (See section 2.2.3.2). SciERC, constructed from scientific literature abstracts, is replete with technical jargon and intricate entity relations, providing a stringent test of our model's capability to process domain-specific language and perform high-precision relation extraction. This dataset is particularly relevant to our research goal of extracting entities and relations from scientific texts, albeit within a different domain.

The NYT dataset, with its array of topics derived from news articles, evaluates the model's ability to generalize across diverse general-domain topics and grasp contextual subtleties across a broad spectrum of subjects. In contrast, the TACRED dataset, which encompasses a wide array of relations and entity types

from newswire and web texts that focus on people, organizations, and locations, tests the effectiveness of the model in recognizing entities and extracting relations within a more structured and formal text, unlike the free-form style typical of news articles.

Together, SciERC assesses our model's proficiency with technical language and complex relationships within a specialized domain, while NYT and TACRED facilitate an assessment of its adaptability and accuracy across a more varied corpus. The selection of these datasets strategically demonstrates the model's generalization capabilities. Strong performance across such diverse datasets would underscore the robustness and adaptability essential for real-world applications.

4.2 Evaluation metrics

We adhere to established evaluation protocols [34], we utilize the F1 measure as our principal evaluation metric for its balanced assessment of precision and recall. The F1 measure is the harmonic mean of precision and recall, providing a single metric that balances both false positives and false negatives, ensuring a comprehensive evaluation of the model's performance. Therefore, it's a fundamental measure for accurately gauging the performance of entity and relation extraction systems.

For the evaluation of the NER model, a predicted entity is only considered correct if it exactly aligns with the annotated data in terms of both span boundaries and entity type. This stringent criterion is crucial as it guarantees high precision in both the detection and classification processes. Such accuracy is imperative for downstream applications, particularly relation extraction, where the validity of relationships often hinges on the correct identification of entity types. Misclassified entities could lead to erroneous or missed relations, thereby compromising the utility of the extracted information.

Regarding the RE model, we used two distinct metrics [23]:
Boundaries Evaluation: This initial, less strict metric qualifies a relation prediction as correct if it accurately identifies the span boundaries of the entities involved and correctly classifies the type of relation. This metric primarily assesses the model's ability to detect and categorize relationships based on their

spatial and contextual presence in the text, without requiring the entity types to be correct. It is particularly useful for preliminary testing of a model's basic relational understanding before more comprehensive assessments.
Strict Evaluation: This more rigorous metric extends the boundaries evaluation by also necessitating the correct classification of entity types involved in the relations. It offers a deeper, more holistic understanding of the text, challenging the model not just to detect and classify relationships accurately but also to ensure precise typing of entities. This metric is crucial for advanced applications where the interplay between entities and their relations critically informs the output, such as in detailed semantic analysis or advanced information retrieval systems. Employing these metrics allows us to meticulously evaluate the nuanced capabilities of our models, ensuring that they not only perform well statistically but also meet the practical demands of real-world applications where precision and reliability are paramount.

4.3 Implementation

In this implementation, we have adapted the PURE entity and relation extraction system originally developed by Zhong et al. [2], available on their GitHub repository [35]. Our objective was to reimplement their system using Jupyter notebooks to enhance the usability and reproducibility of the model's training and evaluation processes. These notebooks are meticulously designed to ensure clarity and ease of execution across different system setups, facilitating straightforward adaptation to new datasets and domains.

The notebooks allow for seamless integration of new Named Entity Recognition (NER) and Relation Extraction (RE) datasets, enabling quick model training and immediate inference on novel data. This approach not only democratizes the accessibility of the PURE system but also sets the stage for future enhancements where new datasets can be effortlessly incorporated.

In this section, we provide a detailed explanation of these Jupyter notebooks. The comprehensive documentation within the notebooks ensures that each step of the model's implementation is clear and easily navigable. For those interested in further exploring the system or replicating our study, the complete implemen-

tation is available through our GitHub repository at PURE: Entity and Relation Extraction from Text System Reproduction [36].

4.3.1 Basic Setup

The initial setup involved configuring our computational environment to meet the requirements for our experiments. This process included installing necessary software dependencies, setting up datasets, and acquiring pre-trained models for NER and RE tasks:

System Specifications:

- Operating System: Windows 11
- Processor: 11th-generation Intel Core i5 CPU
- Memory: 16GB of RAM
- Graphics: Nvidia GeForce RTX 3050 Ti laptop GPU with 4GB of VRAM
- Software: Python 3.1.13 and pip 21.2.2
- Key Library: PyTorch version 1.4.0

Installation Process:

The setup can be carried out using the basic_setup jupyter notebook in our repository. Other than the obvious installation of Python and pip. Other libraries need to be installed. These libraries can be found the requirements.txt file on our repository [36]. To install these requirements the following command can be used:

```
pip install -r requirements.txt
```

Code 4.1: Instalation of project requirements

However, PyTorch version 1.4.0 might be challenging to install. Should any problems arise, it can be installed manually using this command:

```
pip install torch===1.4.0 torchvision===0.5.0 -f https://download.
    pytorch.org/whl/torch_stable.html
```

Code 4.2: Instalation of PyTourch

And if that still doesn't work, another alternative is to install AllenNLP which comes packaged with PyTorch. See Code 4.3 below to install AllenNLP.

```
pip install allennlp-models
```

Code 4.3: Instalation of AllenNLP which includes PyTourch

Data Acquisition: Next, the preprocessed SciERC dataset can be downloaded from their project website [37]. For that, we have implemented some helper functions to download and extract the dataset (see appendix for more detailed code snippets).

Model Preparation: Finally, pre-trained entity and relation models can be downloaded from Princeton University's repository for PURE [38]. These models will be used to reproduce Zhong et al's [2] results that we will build our implementation on.

4.3.2 Model architecture

In our research, we utilized the BERT model as the foundational pre-trained transformer for our NER tasks. The code shown in Code 4.3 exemplifies the initialization of our BERT-based NER model:

```
self.ner_classifier = nn.Sequential(
            FeedForward(input_dim=config.hidden_size * 2 +
    width_embedding_dim,
                        num_layers=2,
                        hidden_dims=head_hidden_dim,
                        activations=F.relu,
                        dropout=0.2),
            nn.Linear(head_hidden_dim, num_ner_labels))
```

Code 4.4: Initialization of the BERT based NER model

Entity Model

Our model employs a feedforward neural network, instantiated through nn.Sequential, comprising two primary layers:

1. FeedForward Layer:

    - **Input Dimension**: The input size is determined by the formula hidden_size * 2 + width_embedding_dim, where the hidden_size parameter represents the doubled hidden size of the BERT model to account for concatenated span embeddings from both start and end embeddings, and width_embedding_dim corresponds to the dimensionality of the width embedding.

- Hidden Layers: There are two hidden layers within the network.
- Hidden Layer Dimensions: Set through the parameter head_hidden_dim (typically 150).
- Activations: Utilizes the Rectified Linear Unit (ReLU) function between layers.
- Dropout: A dropout rate of 20% is applied to regularize the model.

2. Output Layer: Implemented as a linear transformation (nn.Linear) that maps the dimensions from head_hidden_dim to num_ner_labels, where num_ner_labels represents the number of NER labels or classes that the model predicts.

We hope this architecture will allow the model to learn complex representations and dependencies from the annotated training data effectively, thereby improving its ability to generalize across unseen data. The incorporation of dropout and ReLU activation functions is expected to further aid in mitigating the risk of overfitting and enhancing non-linear learning capabilities, respectively.

Relation Model

The relation model is also built upon a pre-trained BERT model to leverage the model's weights and then fine-tune it for our specific task.

The setup of the model includes the following.

- The initialization of the BERT model with the provided configuration.
- A dropout layer with a dropout probability specified.
- A Layer normalization applied to the concatenated output of the subject and object representations.
- A linear layer for classification, mapping the concatenated representation to the output logits.

We must note the setup of the model:

```
self.classifier = nn.Linear(config.hidden_size * 2, self.num_labels)
```

Code 4.5: Initialization of the BERT based RE model

The model is a linear layer that takes the concatenated representation of subject and object entities as input and produces logits for each possible relation label.

- It is defined using nn.Linear and consists of two layers.

- The input dimension is set to the sum of the following:

  hidden_size: corresponds to the hidden size of the BERT model. In BERT, each token in the input sequence is associated with a hidden vector of this size.

  * 2: The * 2 indicates that the representations of the subject and object entities are concatenated. So, the input dimension is twice the hidden size. This is likely because the model wants to capture information from both the start and end embeddings of the entities.

Output Layer:

'self.num_labels' is the number of distinct relation labels the model is designed to classify. Each output neuron in the linear layer corresponds to a specific relation label.

The overall architecture is a simple linear transformation that maps the concatenated representation of subject and object entities to a vector of logits, where each vector element corresponds to the model's prediction for a specific relation label. This linear layer is typically followed by a softmax activation during training to convert the logits into probabilities and compute the cross-entropy loss.

4.3.3 Model training

Entity Model

To establish a baseline, we first attempted to run and evaluate a BERT-based pre-trained entity model [35]. The entity model was run on the SciERC dataset. The outputs, formatted as JSON files where keys represent document and sentence indices and values are lists of predicted entities in the format [start, end, label], serve as inputs for the relation model.

Firstly, the development dataset must be processed. It is loaded into a 'Dataset' object. Then, it is processed using the 'convert_dataset_to_samples' function that transforms the raw dataset into individual samples, each representing a text span with corresponding NER labels. While the 'batchify' function organizes these samples into batches, determined by the eval_batch_size parameter determines the number of samples in each batch. By grouping the samples into batches, we can utilize the computational power of GPUs to evaluate multiple samples simultaneously, significantly speeding up the evaluation process.

```
dev_data = Dataset(dev_data)
```

```
dev_samples, dev_ner = convert_dataset_to_samples(dev_data,
    max_span_length, ner_label2id=ner_label2id, context_window=
    context_window)
dev_batches = batchify(dev_samples, eval_batch_size)
```

Code 4.6: SciERC Development Dataset Processing

Then we can initialize the BERT-based entity model. It is initialized with specific parameters, including the BERT model name ('allenai/scibert_scivocab_uncased'), output directory for saving checkpoints ('bert_model_dir'), and the number of NER labels.

```
bert_model_dir = output_dir
num_ner_labels = len(task_ner_labels[task]) + 1
model = EntityModel(model='allenai/scibert_scivocab_uncased',
    bert_model_dir=bert_model_dir, use_albert=False, max_span_length=
    max_span_length, num_ner_labels=num_ner_labels)
```

Code 4.7: Pre-trained BERT model nitialization

Now that we have the model initialized, we can process the test dataset (similarly to the dev dataset). The model is also evaluated and the NER predictions are saved to a file to be used for training and evaluating the relation model. We will look at the evaluation method in the next section 4.3.4. And then we will review the results in chapter 5.

```
test_data = Dataset(test_data)
prediction_file = os.path.join(output_dir, test_pred_filename)

test_samples, test_ner = convert_dataset_to_samples(test_data,
    max_span_length, ner_label2id=ner_label2id, context_window=
    context_window)
test_batches = batchify(test_samples, eval_batch_size)
evaluate(model, test_batches, test_ner)
output_ner_predictions(model, test_batches, test_data, output_file=
    prediction_file)
```

Code 4.8: Processing the test data and evaluating the pre-trained entity model

Once the baseline was established we trained a BERT-based model from scratch. For this, we trained the pre-trained language models using cross-entropy loss[3.2.3].

$$\mathcal{L}_e = -\sum_{s_i \in S} \log P_e(e_i^* | s_i)$$

A detailed code of the model training can be found in the appendix under the example codes section .1.

Relation Model

Similar to the entity model, a baseline was established using a pre-trained BERT-based model on the SciERC dataset.

The input data for the relation model included sentences, named entities, and predicted named entities from the entity model. The training, development, and test datasets were prepared using the generate_relation_data function.

```
# train set
if do_train:
    train_dataset, train_examples, train_nrel =
    generate_relation_data(train_file, use_gold=True, context_window=
    context_window)
# dev set
if (do_eval and do_train) or (do_eval and not(eval_test)):
    eval_dataset, eval_examples, eval_nrel = generate_relation_data(
    os.path.join(entity_output_dir, entity_predictions_dev), use_gold=
    eval_with_gold, context_window=context_window)
# test set
if eval_test:
    test_dataset, test_examples, test_nrel = generate_relation_data(
    os.path.join(entity_output_dir, entity_predictions_test), use_gold
    =eval_with_gold, context_window=context_window)
```

Code 4.9: Processing dataset fro training the relation model

Then, a BERT-based pre-trained model (allenai/scibert_scivocab_uncased) was used. The configuration parameters included model name, batch sizes, number of epochs, and the learning rate. Key special tokens [CLS] and [SEP] were added to the tokenizer for marking entities and relationships.

Once the baseline was established we trained a BERT-based model from scratch. For this, we trained the pre-trained language models using cross-entropy loss[3.2.3].

$$\mathcal{L}_r = -\sum_{s_i, s_j \in S, s_i \neq s_j} \log P_r(r^*_{i,j} | s_i, s_j)$$

A detailed code of the relation model training can be found in the appendix under the example codes section .1.

4.3.4 Model evaluation

Entity Model

To assess the performance of our entity extraction model, we implemented an evaluation function designed to compute key metrics and provide insight into the model's effectiveness. Below is an in-depth explanation of the evaluation process:

The evaluation function 'evaluate_entity' performs a thorough assessment of

the model using the provided evaluation dataset. This function systematically computes various performance metrics, ensuring a comprehensive evaluation of the model's capabilities.

```
def evaluate_entity(model, batches, tot_gold):
    """
    Evaluate the entity model
    """
    logger.info('Evaluating...')
    c_time = time.time()
    cor = 0
    tot_pred = 0
    l_cor = 0
    l_tot = 0

    for i in range(len(batches)):
        output_dict = model.run_batch(batches[i], training=False)
        pred_ner = output_dict['pred_ner']
        for sample, preds in zip(batches[i], pred_ner):
            for gold, pred in zip(sample['spans_label'], preds):
                l_tot += 1
                if pred == gold:
                    l_cor += 1
                if pred != 0 and gold != 0 and pred == gold:
                    cor += 1
                if pred != 0:
                    tot_pred += 1

    acc = l_cor / l_tot
    logger.info('Accuracy: %5f'%acc)
    logger.info('Cor: %d, Pred TOT: %d, Gold TOT: %d'%(cor, tot_pred,
    tot_gold))
    p = cor / tot_pred if cor > 0 else 0.0
    r = cor / tot_gold if cor > 0 else 0.0
    f1 = 2 * (p * r) / (p + r) if cor > 0 else 0.0
    logger.info('P: %.5f, R: %.5f, F1: %.5f'%(p, r, f1))
    logger.info('Used time: %f'%(time.time()-c_time))
    return f1
```

Code 4.10: entity model evaluation function

The function processes each batch in the evaluation dataset (batches). For each batch, the model generates predictions by running model.run_batch(batches[i], training=False), with the results stored in output_dict['pred_ner'].

For every sample in the batch, the function compares the predicted labels (pred_ner) with the gold standard labels (sample['spans_label']). It increments l_tot for each label comparison. If the prediction matches the gold label, l_cor is incremented. When both the predicted and gold labels are non-zero and match, cor is incremented. Additionally, if the prediction is non-zero, tot_pred is incremented.

Precision (p) is then calculated as the ratio of correct predictions (cor) to the total predictions (tot_pred), defaulting to 0.0 if cor is zero. Recall (r) is determined as the ratio of correct predictions (cor) to the total gold standard labels (tot_gold), also defaulting to 0.0 if cor is zero. The F1 score, representing the harmonic mean of precision and recall, is calculated and defaults to 0.0 if cor is zero.

Relation Model

To assess the performance of our relation extraction model, we implemented an evaluation function designed to compute key metrics and provide insight into the model's effectiveness. Below is an in-depth explanation of the evaluation process:

The evaluation function 'evaluate_relation' performs a thorough assessment of the model using the provided evaluation dataset. This function systematically computes the evaluation loss and various performance metrics, ensuring a comprehensive evaluation of the model's capabilities.

```
def evaluate_relation(model, device, eval_dataloader, eval_label_ids,
    num_labels, e2e_ngold=None, verbose=True):
    model.eval()
    eval_loss = 0
    nb_eval_steps = 0
    preds = []
    for input_ids, input_mask, segment_ids, label_ids, sub_idx,
   obj_idx in eval_dataloader:
        input_ids = input_ids.to(device)
        input_mask = input_mask.to(device)
        segment_ids = segment_ids.to(device)
        label_ids = label_ids.to(device)
        sub_idx = sub_idx.to(device)
        obj_idx = obj_idx.to(device)
        with torch.no_grad():
```

```
14            logits = model(input_ids, segment_ids, input_mask, labels
    =None, sub_idx=sub_idx, obj_idx=obj_idx)
15        loss_fct = CrossEntropyLoss()
16        tmp_eval_loss = loss_fct(logits.view(-1, num_labels),
    label_ids.view(-1))
17        eval_loss += tmp_eval_loss.mean().item()
18        nb_eval_steps += 1
19        if len(preds) == 0:
20            preds.append(logits.detach().cpu().numpy())
21        else:
22            preds[0] = np.append(preds[0], logits.detach().cpu().
    numpy(), axis=0)
23
24    eval_loss = eval_loss / nb_eval_steps
25    logits = preds[0]
26    preds = np.argmax(preds[0], axis=1)
27    result = compute_f1(preds, eval_label_ids.numpy(), e2e_ngold=
    e2e_ngold)
28    result['accuracy'] = simple_accuracy(preds, eval_label_ids.numpy
    ())
29    result['eval_loss'] = eval_loss
30    if verbose:
31        logger.info("***** Eval results *****")
32        for key in sorted(result.keys()):
33            logger.info("  %s = %s", key, str(result[key]))
34    return preds, result, logits
```

Code 4.11: Relation model evaluation function

The model is set to evaluation mode using model.eval(), which disables gradient calculations and certain layers that behave differently during training and evaluation, such as dropout layers. Two variables, eval_loss and nb_eval_steps, are initialized to accumulate the total loss and count the number of evaluation steps, respectively. Additionally, an empty list, preds, is initialized to store predictions.

The function iterates over batches in the evaluation DataLoader (eval_dataloader). For each batch, the input data (input_ids, input_mask, segment_ids, label_ids, sub_idx, obj_idx) is transferred to the specified device (CPU/GPU).

The model computes logits for the input data without calculating gradients, using torch.no_grad(). The cross-entropy loss between the predicted logits and the true labels is then calculated using CrossEntropyLoss. This batch loss is accumulated in eval_loss, and the step count is incremented.

Predicted logits are appended to preds. Once all batches are processed, the total evaluation loss is averaged over the number of evaluation steps to obtain the final evaluation loss. The combined logits from all batches are stored in logits, and the predicted labels are obtained by taking the argmax of these logits along the class dimension.

Finally, the function computes the F1 score using compute_f1[.1], which measures the model's accuracy in identifying the correct relationships.

5 Results

In this chapter we delve into the evaluation results of our model on the SciERC, NYT, and TACRED datasets. We will analyze the model's performance, compare it to the baseline and previous models, and address any observed pitfalls or weaknesses. Finally, we summarize our model's performance across all datasets.

5.1 Dataset statistics

Before we dive into the results, let's have an overview of some of the characteristics of the datasets we've selected. And the implications of this selection on the results.

Table 5.1 shows statistics about the datasets where $|\mathcal{E}|$ is the number of entity types in the dataset, $|\mathcal{R}|$ is the number of relation types, and #sentences is the number of total sentences in each of the training, development, test sets respectively.

5.2 Implications of the chosen datasets

The SciERC (Scientific Information Extraction from Research Papers) dataset is specifically tailored for extracting information from scientific literature. It contains 500 abstracts from AI conference proceedings, annotated for entities, relations, and coreference clusters. The entities are categorized into six types:

Dataset	$\|\mathcal{E}\|$	$\|\mathcal{R}\|$	#Sentences		
			Train	Dev	Test
ScieERC	6	7	350	50	100
NYT	3	24	112392	10000	10000
TACRED	22	42	68124	22631	15509

Table 5.1: The statistics of the datasets. We use SciERC, NYT, and TACRED for evaluating end-to-end relation extraction.

Task, Method, Metric, Material, Other-Scientific-Term, and Generic. This dataset is challenging due to the technical jargon and complex relationships inherent in scientific texts. We chose SciERC to test the model's capability in handling domain-specific language and intricate entity relations.

The New York Times (NYT) dataset is derived from articles published by the New York Times. It includes a vast array of topics, making it suitable for evaluating models across a broad spectrum of subjects. The dataset consists of entities categorized into three types: Location, Person, and Organization, with 24 different relation types. This diversity presents a unique challenge in terms of generalization and contextual understanding. The NYT dataset was selected to assess the model's performance in a real-world, general-domain context.

The TACRED (TAC Relation Extraction Dataset) is one of the largest and most comprehensive relation extraction datasets. It contains over 106,000 sentences drawn from the TAC KBP challenges, annotated with 42 relation types. The entities are categorized into 22 types, including various person and organization subtypes. TACRED is known for its complexity and the variety of relations it encompasses, making it an excellent benchmark for evaluating relation extraction systems. We chose TACRED to test our model's ability to handle diverse and complex relations in a structured format.

5.2.1 Implications of the chosen datasets

Training Stability and Convergence

Larger datasets like NYT and TACRED provide a significant amount of training data, which helps in achieving more stable training and better convergence. With a large number of training examples, the model can learn more diverse

patterns and generalize better to unseen data. Smaller datasets like SciERC, while rich in domain-specific information, might require careful handling to prevent overfitting and to ensure the model learns effectively from limited data.

Model Performance

Larger datasets typically lead to models with higher performance due to the availability of more examples for learning. However, they also demand more computational resources and time for training. Smaller datasets might result in models that are prone to overfitting but can still perform well if the data is of high quality and representative of the target domain.

Generalizability

Larger datasets contribute to better generalizability of the model, enabling it to perform well on a wide range of inputs. The NYT dataset, with its vast number of sentences, allows the model to capture a broad spectrum of linguistic nuances and relations. Conversely, the smaller SciERC dataset focuses on a specialized domain, providing valuable insights but potentially limiting the model's generalizability to other domains.

Resource Requirements

Handling large datasets like NYT and TACRED requires substantial computational resources, including memory and processing power. Efficient data management and preprocessing strategies are essential to ensure that these datasets can be effectively utilized without overwhelming the available resources.

By evaluating our models on these diverse datasets, we aim to demonstrate the generalizability and robustness of our approach across different domains and types of textual data. Each dataset provides unique challenges and opportunities to refine our models, contributing to a comprehensive evaluation of our end-to-end relation extraction system.

5.3 Establishing a baseline

Our model is primarily based on the implementation by Zhong et al. (2020) [2]. To begin, we aimed to replicate their results by running and evaluating

the pre-trained model they provided. This model is based on SciBERT (SciB), a BERT variant pre-trained on scientific texts, and was used on the SciERC dataset.

5.3.1 Pre-trained entity model

Table 5.2 presents the scores obtained from evaluating the pre-trained model on the SciERC dataset for entity extraction.

Model	Dataset	Task	Scores		
			Precision	Recall	F1 Score
Pre-trained model [2]	SciERC	Entity extraction	66.7	66.5	66.6

Table 5.2: Zhong et al's [2] pre-trained model results for entity extraction

The results obtained from running the pre-trained entity model on the SciERC dataset closely match those reported by Zhong et al. (2020) [2]. Our run yielded an F1 score of 66.6, which is very close to the 67.4 reported by Zhong et al. (2020) [2]. This similarity is expected since we are using their pre-trained model.

Furthermore, these results indicate a high degree of reproducibility, suggesting that the pre-trained model is robust and performs consistently under similar conditions. The slight variations in scores can be attributed to differences in the experimental setup, such as hardware configurations, random seeds, or minor variations in data pre-processing.

The model achieved a precision of 66.8, slightly higher than the 66.7 reported by Zhong et al. This indicates that our model is slightly better at correctly identifying entities, though the difference is minimal. Moreove, The recall score of our model is 66.3, compared to 66.5 reported by the original authors. This suggests that our model identified slightly fewer relevant entities, but the difference is again very small, highlighting consistent performance. Leading to the F1 score, which balances precision and recall, is 66.5 in our results and 67.4 in Zhong et al.'s results. This marginal difference demonstrates that our implementation closely matches the original model's performance, validating the reliability of the pre-trained model. The close match between our results and those reported by Zhong et al. (2020) underscores the robustness of the pre-trained entity model. It also validates our experimental setup and the reproducibility of the

model's performance. This consistency is crucial for building confidence in the model's application to other datasets and tasks.

5.3.2 Pre-trained relation model

Table 5.3 presents the scores obtained from evaluating the pre-trained model on the SciERC dataset for relation extraction.

Model	Dataset	Task	Scores		
			Precision	Recall	F1 Score
Pre-trained model [2]	SciERC	Relation extraction	57.92	66.84	62.06

Table 5.3: Zhong et al's [2] pre-trained model results for relation extraction

Our model achieved a precision of 67.10, which is substantially higher than the 57.92 reported by Zhong et al. Precision measures the accuracy of the positive predictions, indicating the proportion of correctly identified relations out of all relations predicted by the model. The higher precision in our results can be attributed to changes in hyperparameters, specifically the batch size, random seed, and learning rate.

Adjusting the batch size might have significantly impacted the model's learning process. A different batch size may have led to more stable gradient updates and better generalization. Furthermore, the choice of random seed affects the initialization of model parameters and the shuffling of data. A different random seed could result in a model that is more accurate in identifying true positive relations. Finally, fine-tuning the learning rate can help the model converge more effectively. A well-chosen learning rate ensures that the model updates its weights in a manner that improves precision. All those factors can explain how the same model performed better on our machine.

The recall score of our model is 68.75, compared to 66.84 reported by the original authors. Recall measures the model's ability to identify all relevant instances in the dataset, indicating the proportion of correctly identified relations out of all actual relations. The slightly higher recall in our results suggests that our model is marginally better at capturing true relations, although the difference is not as pronounced as with precision.

The F1 score is 67.91 in our results and 62.06 in Zhong et al.'s results. This significant difference indicates that our model performs better overall in balancing

the trade-off between precision and recall.

The combined improvements in both precision and recall contribute to a higher F1 score. Our model's ability to reduce false positives while still capturing a high number of true relations leads to this balanced performance. Also, Ensuring consistency in the evaluation procedures and handling edge cases effectively could also have contributed to better F1 scores. The higher scores obtained in our experiments suggest that the pre-trained relation model, when trained and evaluated under our specific conditions, performs better in terms of precision, recall, and F1-score. This indicates that the model is robust and capable of achieving high performance, given the right conditions.

5.4 Evaluation Results Across Datasets

In this section, we present the results obtained from training the SciBERT-based model on each of the three datasets selected for this study. Our analysis highlights the model's performance and provides insights into its effectiveness across different datasets.

5.4.1 SciERC

Entity extraction

The evaluation results of the SciBERT-based model on the SciERC dataset for entity extraction shown in table 5.4 provide a nuanced understanding of the model's performance metrics: a precision of 72.13, a recall of 69.91, and an F1 score of 71.0.

As highlighted before, the SciERC dataset is challenging due to its dense domain-specific terminology and complex sentence structures typical of scientific literature. This complexity affects the model's performance. A precision score of 72.13 indicates that the model accurately identifies entities, minimizing false positives. This high precision reflects the model's effectiveness in handling specialized vocabulary and detailed context within the dataset.

However, a recall score of 69.91 suggests that the model occasionally misses entities, resulting in false negatives.The intricate nature of scientific texts, with specialized terms and context-dependent meanings, likely contributes to this. The model's training may not cover the full diversity of scientific language,

Model	Dataset	Task	Scores		
			Precision	Recall	F1 Score
SciBERT-based	SciERC	Entity extraction	72.13	69.91	71.0

Table 5.4: SciBERT-based model results for entity extraction for the SciERC dataset

leading to gaps in entity recognition. Variability in scientific writing styles and less frequent or emerging scientific terms pose challenges the model needs to address to improve recall. Enhancements could include additional fine-tuning with more diverse scientific texts or integrating supplementary domain-specific resources to broaden the model's understanding and generalization capabilities.

The F1 score of 71.0 indicates that the model accurately identifies entities while capturing most relevant ones. This balance is crucial in practical applications, demonstrating the model's reliability and robustness in real-world scenarios where both precision and recall are important. It shows that the model is not overly biased toward either precision or recall but maintains a commendable equilibrium, ensuring accurate and comprehensive entity predictions.

The use of a specialized encoder significantly contributes to these results. The encoder's ability to process and understand the context-rich and complex language of scientific texts enhances the model's precision by accurately distinguishing entities from non-entities.

Overall, the performance metrics of the SciBERT-based model on the SciERC dataset highlight its strength in handling scientific texts, particularly in terms of precision, while also identifying areas for improving recall. The F1 score underscores the model's balanced capability, making it a reliable tool for entity extraction in the scientific domain.

Relation extraction

The evaluation results of the SciBERT-based model on the SciERC dataset for relation extraction shown in Table 5.5 reveal interesting insights into the model's performance.

The precision score of 71.06 indicates that the model is highly accurate in identifying relations between entities, with relatively few false positives. This high precision reflects the model's strong ability to discern true relationships

within scientific texts, which is crucial for tasks such as information extraction and knowledge graph construction.

However, the recall score of 65.81 reveals that the model misses a significant number of actual relations, failing to recognize approximately 34% of them. This suggests that while the model excels in accuracy, it struggles to capture all instances of relations. The diverse phrasing and complex interdependencies within scientific literature likely contribute to this lower recall.

The F1 score of 68.33 demonstrates a reasonable level of effectiveness, indicating that the model maintains a fair balance between correctly identifying relations and capturing the majority of them. An F1 score in this range shows that while the model is robust, there is room for improvement, particularly in enhancing recall without significantly sacrificing precision.

Model	Dataset	Task	Scores		
			Precision	Recall	F1 Score
SciBERT-based	SciERC	Entity extraction	71.06	65.81	68.33

Table 5.5: SciBERT-based model results for relation extraction for the SciERC dataset

Overall, the performance metrics of the SciBERT-based model on the SciERC dataset highlight its strength in accurately identifying relations within scientific texts. The high precision score indicates a robust ability to identify true relations, while the recall score underscores the need for further enhancements to capture the full range of relevant relations. The balanced F1 score reflects the model's overall reliability and effectiveness in relation extraction within the scientific domain.

5.4.2 NYT

Entity extraction

Moving on to the evaluation results of the SciBERT-based model on the NYT dataset presented in table 5.6.

The NYT dataset, which consists of news articles from The New York Times, poses unique challenges and opportunities for entity extraction. Unlike the dense scientific terminology of the SciERC dataset, the NYT dataset contains a

diverse range of entities across various topics, including people, organizations, locations, and events, presented in more general and varied language. This diversity tests the model's adaptability to different contexts and entity types.

The precision score of 92.86 demonstrates the model's high accuracy in identifying entities within the dataset. This indicates that the model is very effective at minimizing false positives, accurately distinguishing between entities and non-entities. Such a high precision rate is crucial in the context of news articles, where the clarity and correctness of identified entities are vital for reliable information extraction and subsequent applications such as summarization and trend analysis.

The recall score of 89.81 reflects the model's strong capability to capture the vast majority of relevant entities. This high recall rate suggests that the model is adept at recognizing entities across the diverse and varied contexts presented in the NYT dataset, though it still misses a small proportion of entities. The near 90 recall indicates that the model effectively generalizes across the different topics and entity types found in the dataset, ensuring comprehensive entity recognition.

Model	Dataset	Task	Scores		
			Precision	Recall	F1 Score
SciBERT-based	NYT	Entity extraction	92.86	89.81	91.31

Table 5.6: SciBERT-based model results for entity extraction for the NYT dataset

The F1 score of 91.31 signifies that the model maintains an excellent equilibrium between accurately identifying entities and capturing most of the relevant ones. This balance is essential for practical applications, as it ensures that the model is both precise in its entity predictions and comprehensive in its entity coverage. Overall, This score reflects a well-rounded performance, demonstrating that the model does not overly favor either precision or recall but maintains a commendable balance. This equilibrium ensures that the model can be relied upon to perform effectively in real-world scenarios, where both accurate identification and comprehensive recognition of entities are critical.

While the model performs exceptionally well, the slightly lower recall compared to precision suggests there is still room for improvement in capturing every relevant entity. Further fine-tuning with additional diverse news articles

or incorporating more domain-specific training data could enhance the model's ability to recognize even the less frequent or more context-dependent entities, thus improving recall without compromising precision.

The performance metrics of the SciBERT-based model on the NYT dataset underscore its strength in handling diverse and general language contexts, as evidenced by the high precision and recall scores. The F1 score highlights the model's balanced capability, making it a reliable tool for entity extraction in the news domain.

Relation extraction

Table 5.7 presents the scores obtained from evaluating our SciBERT-based model on the NYT dataset for relation extraction.

Model	Dataset	Task	Scores		
			Precision	Recall	F1 Score
SciBERT-based	NYT	Entity extraction	90.13	69.23	86.29

Table 5.7: SciBERT-based model results for relation extraction for the NYT dataset

The precision score of 90.13 indicates that the model excels at accurately identifying relationships between entities, with a low rate of false positives. This high precision is crucial in the context of news articles, where the clarity and correctness of identified relationships are vital for reliable information extraction, enhancing tasks such as summarization, fact-checking, and trend analysis.

However, the recall score of 69.23 highlights a significant challenge: the model misses a notable proportion of actual relationships, failing to recognize approximately 31% of them. This gap suggests that while the model is adept at identifying relations when it does so, it struggles to capture all instances of these relationships. The diverse and complex ways in which relationships are expressed in news articles, including implicit and context-dependent relationships, likely contribute to this lower recall.

The F1 score of 78.04, balancing precision and recall, reflects the model's overall performance. This score signifies that while the model is highly accurate

in identifying relations, there is still substantial room for improvement in capturing the full range of relevant relationships. An F1 score in this range indicates that the model is robust but could benefit from enhancements to improve its comprehensiveness.

To address this, further fine-tuning with a more extensive and varied set of news articles or integrating additional domain-specific resources could help improve the model's recall. Enhancing the model's ability to recognize less common or more subtly expressed relationships would likely raise the recall score, thus improving the F1 score and overall performance.

Overall, the performance metrics of the SciBERT-based model on the NYT dataset highlight its strength in accurately identifying relationships within diverse and general language contexts, as evidenced by the high precision score. The recall score underscores the need for further enhancements to capture the full range of relevant relationships. The F1 score, indicating a balanced capability, underscores the model's overall reliability and effectiveness in relation extraction within the news domain.

5.4.3 TACRED

Entity extraction

Let's now explore our model's performance on the TACRED dataset. The scores are reported in table 5.8 bellow. The evaluation results of the SciBERT-based model on the TACRED dataset for entity extraction offer detailed insights into the model's performance, with a precision of 87.26, a recall of 78.92, and an F1 score of 82.88.

The TACRED dataset, known for its comprehensive and challenging nature, includes a wide range of entity types and relationships, providing a rigorous test for entity extraction models. The precision score of 87.26 indicates that the model is highly accurate in identifying entities, with a low rate of false positives. This high precision reflects the model's ability to correctly distinguish entities within diverse and often complex textual contexts, which is essential for applications requiring precise information extraction.

However, the recall score of 78.92 reveals that the model misses a notable proportion of actual entities, failing to recognize approximately 21 of them. This recall score indicates that while the model is proficient at identifying entities

when it does so, it struggles to capture all relevant entities in the dataset. The diversity and complexity of the entity types in TACRED, along with varied expressions and contexts, likely contribute to this gap in recall.

The F1 score of 82.88, which balances both precision and recall, provides a comprehensive measure of the model's overall performance. An F1 score in this range suggests that the model maintains a good equilibrium between accurately identifying entities and capturing most of them. This balanced performance is crucial for practical applications, where both precision and recall are important to ensure reliability and comprehensiveness in entity extraction.

Model	Dataset	Task	Scores		
			Precision	Recall	F1 Score
SciBERT-based	TACRED	Entity extraction	87.26	78.92	82.88

Table 5.8: SciBERT-based model results for entity extraction for the TACRED dataset

To address this, further fine-tuning with a more extensive and varied set of texts or incorporating additional domain-specific resources could enhance the model's recall. Improving the model's ability to recognize less common or more context-dependent entities would likely raise the recall score, thereby improving the F1 score and overall performance.

Overall, the performance metrics of the SciBERT-based model on the TACRED dataset highlight its strength in accurately identifying entities within diverse and complex textual contexts, as evidenced by the high precision score. The recall score underscores the need for further enhancements to capture the full range of relevant entities. The F1 score, indicating a balanced capability, underscores the model's overall reliability and effectiveness in entity extraction within the varied and challenging contexts of the TACRED dataset.

Relation extraction

Table 5.9 presents the scores obtained from evaluating our SciBERT-based model on the TACRED dataset for relation extraction.

The precision score of 85.71 shows that the model is highly accurate in identifying true relations, with relatively few false positives. This high precision indicates the model's effectiveness in correctly discerning actual relations from

non-relations, which is crucial for ensuring the accuracy and reliability of extracted information. Such precision is essential in applications where incorrect relationship extraction could lead to significant errors in downstream tasks.

The recall score of 98.20 highlights the model's exceptional ability to capture nearly all relevant relations, missing only a small fraction. This high recall suggests that the model is very proficient at recognizing a broad array of relations across different contexts and expressions found in the TACRED dataset, ensuring comprehensive coverage of the data.

Model	Dataset	Task	Scores		
			Precision	Recall	F1 Score
SciBERT-based	TACRED	Entity extraction	85.71	98.20	85.71

Table 5.9: SciBERT-based model results for relation extraction for the TACRED dataset

The F1 score of 91.54 demonstrates the model's strong overall performance by balancing both precision and recall. An F1 score above 90 indicates that the model maintains an excellent equilibrium between accurately identifying relations and capturing the vast majority of them. This balanced performance is crucial for practical applications, ensuring both reliability and comprehensiveness in relation extraction tasks.

Overall, the performance metrics of the SciBERT-based model on the TACRED dataset underscore its strength in accurately identifying and capturing relations within diverse and complex textual contexts. The high precision and recall scores highlight the model's effectiveness, while the F1 score of 91.54 demonstrates a balanced and reliable performance, making the model a robust tool for relation extraction in challenging datasets like TACRED.

5.5 Comparison with previous models

5.5.1 SciERC Dataset

The results of our model on the SciERC dataset are presented in table 5.10. The table compares the F1 scores of our model against several baseline and previous models for both NER and RE tasks.

The models compared in Table 5.1 utilize various encoders. Luan et al., 2018 [9] and 2019 [24] both utilize a combination of LSTM and ELMo (L+E). The use of ELMo provides contextualized word embeddings by considering the entire sentence, while LSTM captures the sequential dependencies within the text. This combination offers a solid performance, achieving F1 scores of 64.2 and 65.2 for entities and 39.3 and 41.6 for relations, respectively.

Wadden et al., 2019 [25] and Zhong et al., 2020 [2] leverage SciBERT (SciB), a BERT variant pre-trained on scientific texts. SciBERT is specifically tuned to handle the technical jargon and context prevalent in scientific literature, thus offering improved performance on the SciERC dataset. Wadden et al. achieved an F1 score of 67.5 for NER and 48.4 for RE, while Zhong et al. slightly outperformed with scores of 67.4 for NER and 50.1 for RE.

Model	Encoder	F1 Score	
		Entity	Relation
Luan et al.,2018 [9]	L+E	64.2	39.3
Luan et al.,2019 [24]	L+E	65.2	41.6
Wadden et al.,2019 [25]	SciB	67.5	48.4
Zhong et al.,2020 [2]	SciB	67.4	50.1
Pre-trained model	SciB	66.6	62.0
Our model	SciB	69.10	68.3

Table 5.10: Test F1 scores on SciERC. The encoders used in different models: L+E = LSTM + ELMo, SciB = SciBERT (size as BERT-base).

The pre-trained model using SciBERT, which we evaluated, shows competitive scores, with an F1 score of 67.4 for NER and a remarkable 62.0 for RE, closely aligning with the scores reported by Zhong et al. in their original paper. This validation confirms the robustness of the pre-trained SciBERT model for extracting entities and relations from scientific texts.

Our model also employs SciBERT, but with additional enhancements and pre-training, which allow it to achieve the highest F1 scores of 69.1 for NER and 68.3 for RE. This significant improvement over other models underscores the advantage of using a specialized encoder pre-trained on domain-specific data combined with further optimization and fine-tuning for the task at hand.

These results highlight the significant impact of the choice of encoders on the performance of entity and relation extraction tasks. Our model, leveraging SciBERT's domain-specific pre-training, has demonstrated superior performance, making it the most effective model for the SciERC dataset to date.

These results also highlight the tremendous advancements made in the field over the past four years. The leap in performance, particularly in relation extraction tasks, underscores the rapid progress in natural language processing techniques and model architectures. Our model's ability to achieve an F1 score of 68.3 in RE compared to the 50.1 F1 score of previous leading models from 2020 demonstrates the efficacy of newer approaches and the importance of continual innovation. This progress not only reflects the enhancements in computational power and data processing but also the growing sophistication of algorithms capable of understanding and interpreting complex scientific texts with unprecedented accuracy.

5.5.2 NYT Dataset

Compared to several recent models, our model's performance on the NYT dataset is summarized in table 5.11. This table presents the F1 scores for both NER and RE tasks, highlighting the efficiency and competitiveness of different approaches.

The NYT dataset results reveal that while our model does not surpass the top scores in relation extraction, it demonstrates competitive performance and significant practical advantages.

Zhao et al., 2021 [39] achieved an F1 score of 90.2 for relation extraction using a BERT-base (Bb) encoder. BERT offers robust contextual embeddings, crucial for understanding and extracting relationships between entities in text.

Huguet et al., 2021 [40] reported notable results with BART, a model that excels in both natural language generation and understanding due to its transformer-based architecture optimized for these tasks. Without pre-training, they achieved an F1 score of 93.1, and with pre-training, they slightly improved the score to 93.4. BART's ability to generate detailed contextual embeddings helps it excel in relation extraction tasks by effectively capturing complex entity relationships.

Tang et al., 2022 [39] set a new benchmark with an F1 score of 93.7 also using BERT. This indicates the effectiveness of BERT in capturing the nuances and contexts of relations in text, thus providing a solid foundation for relation

Model	Encoder	F1 Score	
		Entity	Relation
Zhao et al.,2021 [39]	Bb	-	90.2
Huguet et al.,2021 (no pre-training) [40]	BART	-	93.1
Huguet et al.,2021 [40]	BART	-	93.4
Tang et al.,2022 [39]	Bb	-	93.7
Our model	SciB	91.31	86.29

Table 5.11: Test F1 scores on NYT. The encoders used in different models: Bb = BERT, SciB = SciBERT (size as BERT-base), BART = BART.

extraction tasks.

Our Model employs SciBERT, a BERT variant tailored for scientific texts, which also proves effective for general relation extraction. It achieved an F1 score of 91.31 for entity extraction and 86.29 for relation extraction. Although it does not outperform the state-of-the-art models in relation extraction, it demonstrates respectable performance and offers several advantages over more complex models.

The primary advantages of our model lie in its simplicity and efficiency, which allow for straightforward implementation and maintenance compared to more complex state-of-the-art models. Despite being trained on everyday hardware and requiring only two epochs, our model demonstrates competitive performance across both NER and RE tasks. It offers balanced results and is particularly adaptable to domain-specific content, thanks to the use of SciBERT. This makes it a highly practical and accessible solution for a variety of information extraction tasks, especially in environments with limited computational resources.

The NYT dataset results highlight the trade-offs between the high performance of complex models and the practicality of simpler, efficient solutions like ours. Although our model does not achieve the highest F1 scores for relation extraction, it offers a compelling alternative with its straightforward implementation, efficient training, and balanced performance across tasks. These attributes make it an attractive choice for real-world applications where computational resources and training time are limited, yet reliable performance is required.

5.5.3 TACRED Dataset

The performance of our model on the TACRED dataset, in comparison with several recent models, is summarized in table 5.12. This table presents the F1 scores for both Named Entity Recognition (NER) and Relation Extraction (RE) tasks, providing a comprehensive view of the efficacy of different models.

The comparison of models on the TACRED dataset highlights a variety of approaches and their respective performance metrics:

Zhang et al., 2017 [33] employed an LSTM-based model, which achieved an F1 score of 65.1 for relation extraction. LSTM, known for its capability to capture sequential dependencies in text, offers a solid foundation for extracting relational information. However, the performance is relatively lower compared to more recent models that leverage transformer-based architectures.

Wu et al., 2019 [41] used a BERT-base (Bb) encoder, significantly improving the relation extraction F1 score to 69.4. BERT's transformer architecture, which effectively captures contextual relationships in text, provides a notable performance boost over traditional LSTM models.

Lyu et al., 2021 [42] introduced a model using SpanBERT (SBb), a variant of BERT designed to better capture span-level features and relationships. This approach further enhanced the F1 score to 75.2, showcasing the effectiveness of specialized transformer models in understanding and extracting complex relational information.

Model	Encoder	F1 Score	
		Entity	Relation
Zhang et al.,2017 [33]	LSTM	-	65.1
Wu et al.,2019 [41]	Bb	-	69.4
Lyu et al.,2021 [42]	SBb	-	75.2
Efeoglu et al.,2024 [43]	LLM	-	86.6
Our model	SciB	82.8	85.7

Table 5.12: Test F1 scores on TACRED. The encoders used in different models: Bb = BERT, LSTM = LSTM, SciB = SciBERT (size as BERT-base), SBb = SpanBERT, LLM = LLM.

Efeoglu et al., 2024 [43] leveraged a Large Language Model (LLM), achieving

the highest F1 score of 86.6 for relation extraction. LLMs, with their extensive pre-training on vast amounts of data, excel in capturing intricate relationships and contextual nuances, thereby setting a new benchmark in relation extraction performance.

Our Model utilizes SciBERT, specifically pre-trained on scientific texts, and achieves F1 scores of 82.8 for entity extraction and 85.7 for relation extraction. While it does not surpass the state-of-the-art LLM in relation extraction, it offers robust performance that competes closely with the top-performing models.

The advantages of our model on the TACRED dataset include its balanced performance in both entity recognition and relation extraction, achieving competitive F1 scores of 82.8 and 85.7, respectively. Leveraging SciBERT, it excels in handling domain-specific content, making it adaptable to specialized vocabularies and contexts. The model's simplicity and efficiency, requiring only everyday hardware and minimal training, make it accessible and practical for various applications, particularly in environments with limited computational resources. This combination of robust performance, ease of implementation, and resource efficiency distinguishes our model as a highly versatile and practical solution for comprehensive information extraction tasks.

The TACRED dataset results illustrate the trade-offs between achieving the highest possible F1 scores and maintaining a practical, efficient approach. While the highest-performing model by Efeoglu et al. demonstrates the peak capabilities of LLMs in relation extraction, our model offers a compelling alternative with its balance of strong performance, efficiency, and simplicity. It is well-suited for various real-world applications, particularly in settings where computational resources are constrained, and robust performance across both entity and relation extraction tasks is required.

The comprehensive evaluation across the SciERC, NYT, and TACRED datasets highlights the strengths and practical benefits of our model. It consistently demonstrates competitive performance in both NER and RE tasks, with particularly notable results in the SciERC dataset due to its domain-specific pre-training using SciBERT. Despite the high performance of more complex models, our model strikes a balance between strong performance and practicality, making it an effective choice for a wide range of information extraction tasks where resource efficiency and ease of deployment are crucial. This versatility, combined with robust performance, underscores our model's potential for practical applications in diverse domains.

5.6 Final takeaways

The comprehensive evaluation of our model across the SciERC, NYT, and TACRED datasets highlights several key insights into its performance and robustness. Our model consistently demonstrates strong performance in both Named Entity Recognition (NER) and Relation Extraction (RE) tasks, particularly excelling in domain-specific contexts like the SciERC dataset.

On the SciERC dataset, our model achieved the highest F1 scores of 69.1 for NER and 68.3 for RE, significantly outperforming previous models. This underscores the effectiveness of SciBERT's domain-specific pre-training combined with our additional enhancements and fine-tuning. The results also reflect the substantial advancements in NLP techniques over the past four years, particularly in handling complex scientific texts with improved accuracy and understanding.

This highlights the remarkable advancements made in the field over the past four years. In 2018, models such as those by Luan et al. achieved F1 scores of 64.2 for NER and 39.3 for RE using LSTM and ELMo encoders. Fast forward to 2024, our model, leveraging SciBERT and further enhancements, has significantly pushed these boundaries. This leap in performance underscores the rapid progress in natural language processing techniques, driven by the development of sophisticated transformer-based models like BERT and its variants. These advancements have enabled models to better understand and interpret the complex and nuanced language of scientific texts, leading to unprecedented improvements in accuracy and extraction capabilities.

In the NYT dataset, while our model's F1 score for RE was slightly lower than the state-of-the-art models, it still demonstrated a strong performance with an F1 score of 86.29. The model's high precision of 90.13 indicates its ability to accurately identify relationships, though there is room for improvement in recall. The balanced performance of 91.31 F1 in NER shows the model's adaptability to diverse and general language contexts, making it reliable for real-world applications. These results also highlight the impressive capabilities of BERT-based models in achieving high accuracy across various domains, leveraging their robust contextual embeddings.

Being the largest and most diverse among those we evaluated, the NYT datset posed a significant challenge for our model. This dataset encompasses a vast array of topics and entity types, testing the model's ability to generalize

across varied contexts. Despite achieving a respectable F1 score of 86.29 for relation extraction, our simple model faced a considerable gap when compared to more complex models, such as those leveraging BERT or BART, which scored as high as 93.7. The complexity and diversity of the NYT dataset demand sophisticated architectures capable of capturing intricate patterns and nuances in the data. Our model, while efficient and straightforward, struggles to match the nuanced understanding and comprehensive relational extraction capabilities demonstrated by these advanced models. This performance gap highlights the limitations of simpler approaches in handling the extensive variability present in large-scale, heterogeneous datasets.

For the TACRED dataset, our model achieved competitive F1 scores of 82.8 for NER and 85.7 for RE. While it did not surpass the highest-performing models, it offers robust performance with the advantage of simplicity and efficiency. The high recall score for RE (98.20) highlights the model's capability to capture a broad array of relations, ensuring comprehensive coverage. The consistent results on TACRED further emphasize the strength of BERT variants like SciBERT in extracting detailed relational information from complex datasets.

Overall, our model's balanced performance across all datasets demonstrates its generalizability and robustness. The use of SciBERT, pre-trained on scientific texts, provides a solid foundation for domain-specific tasks, while its efficient training and implementation make it accessible for various practical applications. Despite the competitive landscape of state-of-the-art models, our model offers a compelling alternative with its straightforward approach and strong, consistent results, particularly in environments with limited computational resources. This versatility and reliability make it a valuable tool for information extraction tasks across diverse domains. The strong results obtained with BERT-based models further underscore the ongoing advancements and potential in the field of natural language processing.

6

Conclusions and Future Works

In this thesis, we explored the application of Bidirectional Encoder Representations from Transformers (BERT) for Named Entity Recognition (NER) and Relation Extraction (RE) across various domain-specific datasets, including SciERC, NYT, and TACRED. The primary objective was to evaluate the performance of BERT-based models in handling domain-specific terminologies and complex relational structures, and to compare these results with existing state-of-the-art techniques.

Our results demonstrated that BERT, especially in its SciBERT variant, is highly effective for both NER and RE tasks. The model's ability to capture contextual information bidirectionally significantly improved the accuracy of entity and relation identification. On the SciERC dataset, our model outperformed traditional approaches and matched or exceeded the performance of several contemporary models, achieving F1 scores of 69.1 for entity recognition and 68.3 for relation extraction. This highlights the strength of SciBERT in dealing with technical jargon and complex entity relationships prevalent in scientific literature.

6.1 Summary of Objectives and Achievements

The primary objective of this study was to explore the application of Bidirectional Encoder Representations from Transformers (BERT) for Named Entity Recognition (NER) and Relation Extraction (RE) tasks across various domain-

specific datasets. Specifically, we aimed to evaluate the performance of BERT-based models, including SciBERT, in handling the unique challenges posed by different types of texts and to compare our results with existing state-of-the-art techniques.

In pursuit of this objective, we focused on three distinct datasets: SciERC, NYT, and TACRED. Each dataset presented unique challenges and opportunities for evaluating the capabilities of our models. The SciERC dataset, with its technical jargon and complex relational structures, tested the model's ability to handle specialized scientific texts. The NYT dataset, characterized by its diversity and breadth of topics, required the model to generalize across a wide range of contexts. The TACRED dataset, known for its formal and structured content, allowed us to assess the model's performance in more standardized text environments.

Our study achieved several key milestones:

Outperforming Traditional Approaches on the SciERC Dataset

On the SciERC dataset, our model demonstrated a significant leap in performance over traditional approaches. We achieved an F1 score of 69.1 for entity recognition and 68.3 for relation extraction. These results highlight the strength of SciBERT in dealing with domain-specific terminologies and complex relationships prevalent in scientific literature. By leveraging SciBERT's pre-training on scientific texts, our model could effectively understand and extract relevant entities and their interrelations, outperforming previous models that used older techniques like LSTM and ELMo.

Providing Competitive Results on the NYT Dataset

For the NYT dataset, our model showed strong performance, achieving an F1 score of 91.31 for entity recognition and 86.29 for relation extraction. Although our model did not achieve the highest F1 scores in relation extraction when compared to the most complex contemporary models, it nonetheless provided competitive results. This demonstrates the robustness of our approach and its ability to generalize across different domains without significant performance degradation. The NYT dataset's diversity posed a considerable challenge, yet our model managed to maintain high precision and recall, underscoring its practical applicability in real-world scenarios.

Achieving Competitive Performance on the TACRED Dataset

On the TACRED dataset, our model achieved competitive F1 scores of 82.8 for entity recognition and 85.7 for relation extraction. This performance is notable given the structured nature of the TACRED dataset and the complexity of its relational data. Although our model did not surpass the top-performing large language models (LLMs) in this domain, it provided a balanced performance that highlights its practical utility. The high recall scores indicate the model's capability to capture a broad array of relations, essential for comprehensive information extraction tasks. Overall, the achievements of our study underscore the efficacy of BERT-based models, particularly SciBERT, in enhancing the performance of NER and RE tasks across varied domains. Our results validate that pre-trained models like BERT and SciBERT offer significant advantages, not only matching but often exceeding the performance of more traditional approaches, all while requiring lower computational overhead. These findings emphasize the potential of BERT-based models for a wide range of applications, making them accessible and practical tools for diverse information extraction tasks.

6.2 Summary of the results

SciERC Dataset

The SciERC dataset posed several specific challenges that tested the capabilities of our BERT-based models, particularly SciBERT. This dataset, derived from scientific literature, includes dense technical jargon and intricate relationships between entities, which are not typically encountered in more general text corpora.

Challenges of the SciERC Dataset:

1. Technical Jargon: The SciERC dataset consists of abstracts from AI conference proceedings, containing specialized terminology unique to the field of artificial intelligence. This technical language can be difficult for models trained on general corpora to understand and process accurately.

2. Complex Relationships: The dataset requires the extraction of complex relationships between entities such as methods, metrics, materials, tasks, and other scientific terms. These relationships are often deeply contextual and can be challenging to identify without a robust understanding of the underlying scientific concepts.

3. Sparse and Varied Data: Given the specialized nature of the dataset, the number of instances for certain entities and relationships is relatively sparse, making it harder for models to learn and generalize these patterns effectively.

Advantages of SciBERT's Pre-training: SciBERT, a variant of BERT pre-trained on a large corpus of scientific texts, provided a significant advantage in addressing these challenges. The pre-training on domain-specific literature equipped SciBERT with a deep understanding of scientific terminology and context, allowing it to more accurately capture and represent the nuanced meanings of technical terms and the relationships between them.

SciBERT's pre-training involved extensive exposure to scientific articles, enabling it to develop a rich contextual understanding of scientific language. This contextual knowledge was crucial in identifying and classifying entities accurately within the complex sentences typical of scientific abstracts. The model's ability to grasp the intricate relationships between entities was significantly enhanced by its pre-training on scientific texts. This pre-training allowed the model to recognize and extract relationships that are highly context-dependent, which is a common characteristic of scientific literature.

When comparing the performance of our SciBERT-based model to previous models, the improvements are notable and highlight the effectiveness of our approach:

Our model achieved F1 scores of 69.1 for entity recognition and 68.3 for relation extraction. In contrast, earlier models such as those by Luan et al. (2018) using LSTM and ELMo achieved F1 scores of 64.2 for entity recognition and 39.3 for relation extraction. This significant improvement underscores the superior capability of SciBERT in handling scientific texts. Another benchmark, the model by Zhong et al. (2020) using SciBERT, achieved F1 scores of 66.6 for entity recognition and 50.1 for relation extraction. Our enhancements and fine-tuning pushed these scores higher, demonstrating our model's optimized performance.

The reduction in errors, particularly false negatives, is a critical improvement. By better capturing the nuances of scientific terminology and relationships, our model was able to significantly decrease the number of missed entities and relations, leading to higher recall rates. The precision of the model also improved, indicating fewer false positives and more accurate predictions, which is crucial in applications where precision is paramount.

Previous models struggled with the specific jargon and complex relationships found in scientific literature. Our SciBERT-based approach showed a marked improvement in adapting to and processing this specialized language, validating the importance of domain-specific pre-training. The ability to handle sparse data effectively, which is often a challenge in specialized domains, was another area where our model outperformed previous methods.

In summary, the SciERC dataset results highlight the substantial advancements made possible through the use of SciBERT. By leveraging pre-training on scientific texts, our model achieved notable improvements in both entity recognition and relation extraction, setting a new benchmark for performance in this challenging domain. The comparison with previous models underscores the effectiveness of our approach and the critical role of domain-specific pre-training in achieving high accuracy and robust performance.

NYT Dataset

Diversity and Complexity of the NYT Dataset: The New York Times (NYT) dataset is one of the most extensive and diverse collections used for Named Entity Recognition (NER) and Relation Extraction (RE) tasks. This dataset includes a vast array of articles covering numerous topics such as politics, economics, sports, science, and culture. The wide range of topics results in a rich diversity of entity types, including but not limited to persons, organizations, locations, events, and products. Each topic brings its unique context and terminology, presenting a significant challenge for models tasked with accurately identifying and extracting entities and their relationships.

1. Wide Range of Entity Types: The dataset contains entities that vary significantly in terms of frequency and context. For instance, while common entities like well-known persons or major cities appear frequently, other entities like specific events or lesser-known organizations might appear only sporadically.

2. Complex Contextual Variations: Articles in the NYT dataset are written in diverse styles and formats, including news reports, opinion pieces, and feature stories. Each format uses language differently, adding another layer of complexity to the task of entity recognition and relation extraction.

3. Ambiguous and Context-Dependent Entities: The same word or phrase can represent different entities depending on the context. For example, "Washington" could refer to a person, a city, or even a sports team. Disambiguating these entities requires a model to understand and leverage the surrounding context effectively.

Why More Complex Models Performed Better:

The performance gap between simpler models like ours and more complex state-of-the-art models can be attributed to several architectural and training-related factors:

1. dvanced Architectures:

    - Transformer Models: More complex models often use advanced transformer architectures like BERT, RoBERTa, or BART, which are designed to handle large-scale language understanding tasks. These models leverage multiple attention heads and layers to capture long-range dependencies and nuanced contextual information more effectively than simpler models.
    - Pre-trained Language Models: These models benefit from extensive pre-training on massive and diverse corpora, enabling them to generalize well across different domains. For instance, BART's architecture, optimized for both generation and comprehension, allows it to excel in extracting detailed contextual embeddings that are crucial for accurate relation extraction.

2. Training Techniques:

    - Fine-Tuning on Diverse Data: Complex models often undergo rigorous fine-tuning on domain-specific data, enhancing their ability to handle the specific characteristics of each dataset. This targeted fine-tuning improves their performance on tasks involving diverse topics and entity types.
    - Data Augmentation and Regularization: Techniques like data augmentation and dropout regularization help complex models avoid overfitting and improve generalization. By artificially expanding the training set and introducing noise during training, these methods ensure that the models remain robust across different contexts.

3. Handling Ambiguity and Context-Dependence:

    - Contextual Embeddings: Advanced models use contextual embeddings to dynamically adjust the representation of words based on their surrounding context. This ability is crucial for disambiguating entities and accurately identifying relationships in varied and complex articles.
    - Attention Mechanisms: The attention mechanisms in transformer models allow them to focus on relevant parts of the text, improving their ability to capture subtle relational nuances and context-dependent meanings.

Robustness and Generalizability of Our Model: Despite the simpler architecture, our model demonstrated a notable degree of robustness and generalizability on the NYT dataset:

1. Competitive Results: Our model achieved an F1 score of 91.31 for entity recognition and 86.29 for relation extraction, which, while not the highest, are still competitive. These results highlight the model's ability to generalize across different topics and contexts, performing reliably across the diverse articles in the dataset.
2. Simplicity and Efficiency: The simplicity of our model, requiring minimal computational resources, makes it a practical choice for real-world applications where powerful hardware may not be available. This efficiency did not come at the cost of significant performance degradation, demonstrating that simpler models can still offer robust solutions.
3. High Precision: Our model maintained a high precision score, indicating its effectiveness in accurately identifying entities and relationships without producing many false positives. This attribute is particularly valuable in applications where the cost of errors is high.

In summary, while more complex models outperform in handling the diverse and complex nature of the NYT dataset due to their advanced architectures and sophisticated training techniques, our simpler model still showcased robustness and generalizability. Its competitive performance, combined with efficiency and practical applicability, underscores the potential of simpler architectures in delivering reliable results, especially in resource-constrained environments.

TACRED Dataset

The TACRED (TAC Relation Extraction Dataset) is one of the largest and most comprehensive datasets for relation extraction tasks, known for its formal and highly structured content. The dataset includes over 106,000 sentences drawn from newswire and web text, annotated with 42 relation types between named entities. Entities are categorized into 22 types, including various person and organization subtypes. The sentences in TACRED are typically well-formed, grammatically correct, and contain clearly defined relationships, which makes it an excellent benchmark for evaluating the effectiveness of relation extraction models.

1. High-Quality Annotations: The dataset provides high-quality, manually annotated relationships, ensuring reliable ground truth data for training and evaluation.

2. Diverse Entity Types and Relations: With 22 entity types and 42 relation types, TACRED covers a wide range of relational patterns and contexts, providing a comprehensive test bed for model evaluation.
3. Consistent Sentence Structure: The formal nature of the sentences, drawn from professional sources, allows models to leverage grammatical and syntactical cues more effectively than in less structured datasets.

Our model's performance on the TACRED dataset was competitive, achieving F1 scores of 82.8 for entity recognition and 85.7 for relation extraction. When compared to other state-of-the-art models, several trade-offs and observations emerge:

1. Alignment with State-of-the-Art:
    - Our model's F1 score of 85.7 for relation extraction is commendable, particularly in light of the high-performing models like those by Efeoglu et al. (2024), which achieved an F1 score of 86.6. While our model did not surpass the top-performing models, it demonstrated robust performance that is closely aligned with the leading benchmarks.
    - The performance gap, though present, is relatively narrow, highlighting that our approach remains competitive despite being simpler and less resource-intensive than the latest large language models (LLMs).

2. Trade-offs Involved:
    - Computational Efficiency: One of the key trade-offs with our model is its efficiency. Unlike more complex models that require extensive computational resources and training time, our model was trained on everyday hardware with minimal resources. This makes our approach more accessible and practical for real-world applications where resource constraints are a consideration.
    - Simplicity vs. Performance: The slight performance trade-off, seen in the marginally lower F1 scores, is balanced by the model's simplicity and ease of deployment. For many applications, this trade-off is acceptable, especially when the slight dip in accuracy does not critically impact the overall utility of the system.

The formal and structured nature of the TACRED dataset provided a rigorous testing ground for our model. Our model demonstrated strong, competitive performance, aligning closely with state-of-the-art results while maintaining a balance between efficiency and accuracy. Despite some weaknesses, particularly in precision and handling complex relationships, the strengths observed highlight the model's robustness and adaptability, making it a viable option for a range of information extraction tasks.

Significance of Findings

The findings from our study have several important implications for the field of natural language processing (NLP). Our results demonstrate the efficacy of BERT-based models, particularly SciBERT, in handling complex tasks such as Named Entity Recognition (NER) and Relation Extraction (RE) across diverse datasets. This underscores the transformative impact of pre-trained language models on NLP tasks, marking a significant advancement over traditional approaches. The ability of these models to capture deep contextual information and handle nuanced language elements positions them as crucial tools for future NLP research and applications.

Pre-trained models like BERT and SciBERT have proven to be game-changers in NLP due to their ability to achieve high performance with relatively lower computational overhead compared to traditional methods and even some more complex contemporary models. The key advantages include:

1. Robust Contextual Understanding:

    - BERT's bidirectional training allows it to consider both preceding and following contexts in a text, leading to a more nuanced understanding of language. This is particularly beneficial for tasks like NER and RE, where context is crucial for accurate identification and extraction.
    - SciBERT, pre-trained on scientific literature, further enhances this capability by incorporating domain-specific knowledge, making it particularly effective for technical and specialized texts.

2. Efficient Transfer Learning: The transfer learning approach utilized by BERT and SciBERT allows these models to be fine-tuned on specific tasks with relatively small datasets, reducing the need for extensive task-specific data collection and annotation. This efficiency in transfer learning is a significant step forward, enabling rapid development and deployment of NLP solutions across various domains.
3. Lower Computational Overhead: Despite their sophisticated architecture, BERT-based models can be fine-tuned with modest computational resources. This contrasts with the extensive resources typically required for training complex neural networks from scratch, making these models more accessible for a wide range of applications.

The practical applications of our model are extensive and varied, underscoring its efficiency and simplicity. Its ability to accurately recognize entities and extract relationships from text makes it valuable for information extraction tasks in healthcare, finance, legal, and academic research. For instance, it can

extract patient information from clinical notes to aid in patient management and research. The model's efficiency and lower computational requirements make it suitable for resource-constrained environments, democratizing access to advanced NLP capabilities for small businesses, non-profits, and research institutions. Its simple architecture ensures easy implementation and maintenance, reducing barriers to entry for organizations adopting NLP solutions. The model's adaptability to different domains, as demonstrated across the SciERC, NYT, and TACRED datasets, highlights its versatility for domain-specific customization in academic research tools, search engines, and content analysis platforms. Additionally, by providing accurate information extraction, the model enhances decision-making processes in various fields, from organizing legal case information to analyzing market trends in business intelligence.

The significance of our findings lies in the validation of pre-trained models like BERT and SciBERT as powerful tools for NER and RE tasks. Their ability to deliver high performance with lower computational demands opens up numerous practical applications, particularly in settings where efficiency and simplicity are paramount. The broader implications for NLP are profound, as these models set new benchmarks for what can be achieved with relatively modest resources, paving the way for more widespread adoption and innovation in the field.

6.3 Future works

In this section, we outline several potential directions for extending and enhancing the research presented in this thesis. The primary areas for future work include improving computational resources, data augmentation, model architecture enhancements, and optimization techniques.

Hardware Upgrades

Utilizing more powerful computational resources, such as advanced GPUs or TPUs, can significantly enhance the efficiency and effectiveness of training deep learning models. The current study relied on available hardware, which may have limited the complexity and scale of the model training. Upgrading to GPUs with higher memory and faster processing capabilities, or leveraging TPUs designed specifically for training large-scale neural networks, can provide

several benefits:

- Speed: Higher computational power can drastically reduce training times, allowing for more extensive experimentation and hyperparameter tuning.
  - Extensive Experimentation: Faster training times enable researchers to conduct more experiments within the same timeframe, allowing for the exploration of a wider range of model architectures, hyperparameter settings, and training strategies. This iterative process is crucial for refining models and achieving optimal performance.
  - Rapid Prototyping: Quick iterations can lead to faster prototyping and validation of new ideas, facilitating the identification of the most promising approaches without the long waiting periods associated with slower hardware.
- Batch Size: Larger batch sizes can be used without running into memory limitations, which can improve the stability of the training process and potentially lead to better model convergence.
  - Improved Stability: Larger batch sizes can reduce the variance in gradient estimates, leading to a smoother and more stable training process. This stability can help the model converge more reliably and efficiently.
  - Enhanced Convergence: With the ability to process more data in each training step, larger batch sizes can lead to faster convergence. This is particularly beneficial when working with large datasets, as it can shorten the overall training duration while achieving comparable or better model performance.

Extended Training

Increasing the number of training epochs can further refine the model's learning process, particularly for complex and large-scale datasets such as SciERC, NYT, and TACRED. Extended training can help the model to:

- Better Convergence: Longer training allows the model to achieve better convergence, reducing the risk of underfitting. This is especially important for capturing nuanced relationships and rare entities within the data.
- Pattern Recognition: Over more epochs, the model can learn more subtle patterns and dependencies in the data, which can improve its performance on both Named Entity Recognition (NER) and Relation Extraction (RE) tasks.

- Generalization: Extended training with appropriate regularization can help improve the model's generalization capabilities, making it more robust to unseen data.

In implementing these enhancements, it is crucial to monitor the training process to avoid overfitting, where the model becomes too specialized to the training data and performs poorly on new data. Techniques such as early stopping, learning rate scheduling, and regularization methods can be employed to mitigate this risk.

By leveraging more powerful hardware and extending the training duration, future research can push the boundaries of current model performance, leading to more accurate and reliable NER and RE systems. These improvements are expected to significantly impact the ability of such models to handle domain-specific terminologies and complex relational structures, thus enhancing their applicability across various contexts and datasets.

This thesis has presented an in-depth exploration of the application of Bidirectional Encoder Representations from Transformers (BERT) for Named Entity Recognition (NER) and Relation Extraction (RE). The primary objectives were to evaluate the performance of BERT-based models on domain-specific datasets, compare them with existing state-of-the-art techniques, and develop a framework for efficient training and application of these models across various contexts.

Throughout this research, we have demonstrated that BERT-based models are capable of performing on par with, and in some cases surpassing, current state-of-the-art models in terms of efficiency and computational overhead. Our experiments across diverse datasets, including SciERC, NYT, and TACRED, have underscored the versatility and robustness of BERT in handling complex and domain-specific language structures.

Several key findings have emerged from this study:

- Model Performance: BERT-based models show strong performance in both NER and RE tasks, with significant potential for further improvement through hardware upgrades, extended training, and data augmentation.
- Efficiency: The efficiency of BERT-based models in terms of computational resources makes them an attractive option for practical applications where minimizing resource use is crucial.
- Future Enhancements: Potential improvements through advanced hardware, extensive training, data augmentation, and the inclusion of more diverse datasets highlight the avenues for future research. The research

also identified several challenges and limitations, such as the need for more powerful computational resources and the importance of balancing training time with model performance. Addressing these challenges will be critical for advancing the capabilities of BERT-based models and ensuring their practical applicability.

In closing, this thesis contributes to the ongoing development of natural language processing techniques by providing a comprehensive evaluation of BERT for NER and RE tasks. The findings offer valuable insights for both academic research and practical applications, highlighting the potential for BERT-based models to transform the way we extract and utilize information from textual data.

Future research can build on this work by exploring the proposed enhancements and addressing the identified challenges. By continuing to refine and expand the capabilities of BERT-based models, we can move closer to achieving more accurate, efficient, and versatile information extraction systems. This will have a profound impact on a wide range of applications, from knowledge base construction and information retrieval to automated question-answering and beyond.

References

[1] Devlin, J. et al. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. 2019. arXiv: 1810.04805 [cs.CL]. URL: https://arxiv.org/abs/1810.04805.

[2] Zhong, Z. and Chen, D. "A Frustratingly Easy Approach for Joint Entity and Relation Extraction". In: *ArXiv* abs/2010.12812 (2020). URL: https://api.semanticscholar.org/CorpusID:232320859.

[3] Goyal, A., Gupta, V., and Kumar, M. "Recent Named Entity Recognition and Classification techniques: A systematic review". In: *Computer Science Review* 29 (Aug. 2018), pp. 21–43.

[4] Fang, Z. et al. "TEBNER: Domain Specific Named Entity Recognition with Type Expanded Boundary-aware Network". In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Ed. by Moens, M.-F. et al. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 198–207. URL: https://aclanthology.org/2021.emnlp-main.18.

[5] Bhandari, N. et al. "Resolving Ambiguities in Named Entity Recognition Using Machine Learning". In: *2017 International Conference on Next Generation Computing and Information Systems (ICNGCIS)*. 2017, pp. 159–163.

[6] Tsai, C.-T., Mayhew, S., and Roth, D. "Cross-Lingual Named Entity Recognition via Wikification". In: *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*. Ed. by Riezler, S. and Goldberg, Y. Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 219–228. URL: https://aclanthology.org/K16-1022.

[7] Le, P. and Titov, I. *Improving Entity Linking by Modeling Latent Relations between Mentions*. 2018. arXiv: 1804.10637 [cs.CL].

[8] Linguistic Data Consortium, T.T.o.t.U.o.P. *ACE*. https://www.ldc.upenn.edu/collaborations/past-projects/ace.

[9] Luan, Y. et al. "Multi-Task Identification of Entities, Relations, and Coreferencefor Scientific Knowledge Graph Construction". In: *Proc. Conf. Empirical Methods Natural Language Process. (EMNLP)*. 2018.

[10] Tripathi, S. *New York Times Relation Extraction Dataset*. https://www.kaggle.com/datasets/daishinkan002/new-york-times-relation-extraction-dataset/data.

[11] Zhong Victor, e.a. *TAC Relation Extraction Dataset LDC2018T24*. https://catalog.ldc.upenn.edu/LDC2018T24. Dec. 2018.

[12] Li, Q. and Ji, H. "Incremental Joint Extraction of Entity Mentions and Relations". In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Toutanova, K. and Wu, H. Baltimore, Maryland: Association for Computational Linguistics, June 2014, pp. 402–412. URL: https://aclanthology.org/P14-1038.

[13] Zhang, M., Zhang, Y., and Fu, G. "End-to-End Neural Relation Extraction with Global Optimization". In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Ed. by Palmer, M., Hwa, R., and Riedel, S. Copenhagen, Denmark: Association for Computational Linguistics, Sept. 2017, pp. 1730–1740. URL: https://aclanthology.org/D17-1182.

[14] Wang, J. and Lu, W. "Two are Better than One: Joint Entity and Relation Extraction with Table-Sequence Encoders". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Ed. by Webber, B. et al. Online: Association for Computational Linguistics, Nov. 2020, pp. 1706–1721. URL: https://aclanthology.org/2020.emnlp-main.133.

[15] Miwa, M. and Sasaki, Y. "Modeling Joint Entity and Relation Extraction with Table Representation". In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Ed. by Moschitti, A., Pang, B., and Daelemans, W. Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1858–1869. URL: https://aclanthology.org/D14-1200.

[16] Devlin, J. et al. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. 2019. arXiv: 1810.04805 [cs.CL].

[17] Katiyar, A. and Cardie, C. "Going out on a limb: Joint Extraction of Entity Mentions and Relations without Dependency Trees". In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Barzilay, R. and Kan, M.-Y. Vancouver, Canada: Association for Computational Linguistics, July 2017, pp. 917–928. URL: https://aclanthology.org/P17-1085.

[18] Zheng, S. et al. "Joint Extraction of Entities and Relations Based on a Novel Tagging Scheme". In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Barzilay, R. and Kan, M.-Y. Vancouver, Canada: Association for Computational Linguistics, July 2017, pp. 1227–1236. URL: https://aclanthology.org/P17-1113.

[19] Sun, C. et al. "Joint Type Inference on Entities and Relations via Graph Convolutional Networks". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Ed. by Korhonen, A., Traum, D., and Màrquez, L. Florence, Italy: Association for Computational Linguistics, July 2019, pp. 1361–1370. URL: https://aclanthology.org/P19-1131.

[20] Fu, T.-J., Li, P.-H., and Ma, W.-Y. "GraphRel: Modeling Text as Relational Graphs for Joint Entity and Relation Extraction". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Ed. by Korhonen, A., Traum, D., and Màrquez, L. Florence, Italy: Association for Computational Linguistics, July 2019, pp. 1409–1418. URL: https://aclanthology.org/P19-1136.

[21] Li, X. et al. "Entity-Relation Extraction as Multi-Turn Question Answering". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Ed. by Korhonen, A., Traum, D., and Màrquez, L. Florence, Italy: Association for Computational Linguistics, July 2019, pp. 1340–1350. URL: https://aclanthology.org/P19-1129.

[22] Miwa, M. and Bansal, M. "End-to-End Relation Extraction using LSTMs on Sequences and Tree Structures". In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Erk, K. and Smith, N.A. Berlin, Germany: Association for Computational

Linguistics, Aug. 2016, pp. 1105–1116. URL: https://aclanthology.org/P16-1105.

[23] Bekoulis, G. et al. "Adversarial training for multi-context joint entity and relation extraction". In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Ed. by Riloff, E. et al. Brussels, Belgium: Association for Computational Linguistics, Oct. 2018, pp. 2830–2836. URL: https://aclanthology.org/D18-1307.

[24] Luan, Y. et al. "A general framework for information extraction using dynamic span graphs". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Ed. by Burstein, J., Doran, C., and Solorio, T. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 3036–3046. URL: https://aclanthology.org/N19-1308.

[25] Wadden, D. et al. "Entity, Relation, and Event Extraction with Contextualized Span Representations". In: *ArXiv* abs/1909.03546 (2019). URL: https://api.semanticscholar.org/CorpusID:202539496.

[26] Lee, K. et al. "End-to-end Neural Coreference Resolution". In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Ed. by Palmer, M., Hwa, R., and Riedel, S. Copenhagen, Denmark: Association for Computational Linguistics, Sept. 2017, pp. 188–197. URL: https://aclanthology.org/D17-1018.

[27] He, L. et al. "Jointly Predicting Predicates and Arguments in Neural Semantic Role Labeling". In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Ed. by Gurevych, I. and Miyao, Y. Melbourne, Australia: Association for Computational Linguistics, July 2018, pp. 364–369. URL: https://aclanthology.org/P18-2058.

[28] Lin, Y. et al. "A Joint Neural Model for Information Extraction with Global Features". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Ed. by Jurafsky, D. et al. Online: Association for Computational Linguistics, July 2020, pp. 7999–8009. URL: https://aclanthology.org/2020.acl-main.713.

[29] Bebis, G. and Georgiopoulos, M. "Feed-forward neural networks". In: *IEEE Potentials* 13.4 (1994), pp. 27–31.

[30] Zhang, Z. et al. "ERNIE: Enhanced Language Representation with Informative Entities". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Ed. by Korhonen, A., Traum, D., and Màrquez, L. Florence, Italy: Association for Computational Linguistics, July 2019, pp. 1441–1451. URL: https://aclanthology.org/P19-1139.

[31] Baldini Soares, L. et al. "Matching the Blanks: Distributional Similarity for Relation Learning". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Ed. by Korhonen, A., Traum, D., and Màrquez, L. Florence, Italy: Association for Computational Linguistics, July 2019, pp. 2895–2905. URL: https://aclanthology.org/P19-1279.

[32] Peters, M.E. et al. *Knowledge Enhanced Contextual Word Representations*. 2019. arXiv: 1909.04164 [cs.CL].

[33] Zhang, Y. et al. "Position-aware Attention and Supervised Data Improve Slot Filling". In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Ed. by Palmer, M., Hwa, R., and Riedel, S. Copenhagen, Denmark: Association for Computational Linguistics, Sept. 2017, pp. 35–45. URL: https://aclanthology.org/D17-1004.

[34] Taillé, B. et al. "Let's Stop Incorrect Comparisons in End-to-end Relation Extraction!" In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Ed. by Webber, B. et al. Online: Association for Computational Linguistics, Nov. 2020, pp. 3689–3701. URL: https://aclanthology.org/2020.emnlp-main.301.

[35] *PURE: Entity and Relation Extraction from Text*. https://github.com/princeton-nlp/PURE. Accessed: 2023-10-07.

[36] *PURE: Entity and Relation Extraction from Text system reproduction*. https://github.com/OdaiMohammad/pure-ere-reproduction. Accessed: 2023-10-07.

[37] *Multi-Task Identification of Entities, Relations, and Coreferencefor Scientific Knowledge Graph Construction*. https://nlp.cs.washington.edu/sciIE/. Accessed: 2023-10-07.

[38] *Princeton PURE project*. https://nlp.cs.princeton.edu/projects/pure/scierc_models/. Accessed: 2023-10-07.

[39] Tang, W. et al. "UniRel: Unified Representation and Interaction for Joint Relational Triple Extraction". In: *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. Ed. by Goldberg, Y., Kozareva, Z., and Zhang, Y. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, Dec. 2022, pp. 7087–7099. URL: https://aclanthology.org/2022.emnlp-main.477.

[40] Huguet Cabot, P.-L. and Navigli, R. "REBEL: Relation Extraction By End-to-end Language generation". In: *Findings of the Association for Computational Linguistics: EMNLP 2021*. Ed. by Moens, M.-F. et al. Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 2370–2381. URL: https://aclanthology.org/2021.findings-emnlp.204.

[41] Wu, S. and He, Y. *Enriching Pre-trained Language Model with Entity Information for Relation Classification*. 2019. arXiv: 1905.08284 [cs.CL].

[42] Lyu, S. and Chen, H. *Relation Classification with Entity Type Restriction*. 2021. arXiv: 2105.08393 [cs.CL].

[43] Efeoglu, S. and Paschke, A. *Retrieval-Augmented Generation-based Relation Extraction*. 2024. arXiv: 2404.13397 [cs.CL].

Appendix

.1 Example codes developed for the project

```
    def download_file(file_name, url):
        file_name = os.getcwd() + file_name
        os.makedirs(os.path.dirname(file_name), exist_ok=True)
        r = requests.get(url, stream=True)

        r.raise_for_status()

        with open(file_name, 'wb') as f:
            pbar = tqdm(unit="B", total=int(r.headers['Content-Length
'']), position=0, leave=True, desc='Downloading')
            for chunk in r.iter_content(chunk_size=1024):
                if chunk: # filter out keep-alive new chunks
                    pbar.update(len(chunk))
                    f.write(chunk)

    def extract_tar_file(file_name, target_directory):
        with tarfile.open(name=os.getcwd() + file_name) as tar:
            for member in tqdm(iterable=tar.getmembers(), total=len(
tar.getmembers()), desc='Extracting'):
                tar.extract(member=member, path=os.getcwd() +
target_directory)

    def unzip_file(file_name, target_directory):
        with zipfile.ZipFile(os.getcwd() + file_name) as zf:
            for member in tqdm(zf.infolist(), desc='Extracting'):
                zf.extract(member, os.getcwd() + target_directory)
```

Code 1: Helper functions used to download and extract files

```
    for _ in tqdm(range(num_epoch), position=0, leave=True):
        if train_shuffle:
            random.shuffle(train_batches)
        for i in tqdm(range(len(train_batches)), position=0, leave=
    True):
            output_dict = model.run_batch(train_batches[i], training=
    True)
            loss = output_dict['ner_loss']
            loss.backward()

            tr_loss += loss.item()
            tr_examples += len(train_batches[i])
            global_step += 1

            optimizer.step()
            scheduler.step()
            optimizer.zero_grad()

            if global_step % print_loss_step == 0:
                logger.info('Epoch=%d, iter=%d, loss=%.5f'%(_, i,
    tr_loss / tr_examples))
                tr_loss = 0
                tr_examples = 0

            if global_step % eval_step == 0:
                f1 = evaluate(model, dev_batches, dev_ner)
                if f1 > best_result:
                    best_result = f1
                    logger.info('!!! Best valid (epoch=%d): %.2f' % (
    _, f1*100))
                    save_model(model, output_dir)
```

Code 2: Training the entity model from scratch

```
    train_features = convert_examples_to_features(
    train_examples, label2id, max_seq_length, tokenizer,
    special_tokens, unused_tokens=not(add_new_tokens))
if train_mode == 'sorted' or train_mode == 'random_sorted':
    train_features = sorted(train_features, key=lambda f: np.sum(f.
    input_mask))
else:
    random.shuffle(train_features)
all_input_ids = torch.tensor([f.input_ids for f in train_features],
    dtype=torch.long)
```

```
all_input_mask = torch.tensor([f.input_mask for f in train_features], dtype=torch.long)
all_segment_ids = torch.tensor([f.segment_ids for f in train_features], dtype=torch.long)
all_label_ids = torch.tensor([f.label_id for f in train_features], dtype=torch.long)
all_sub_idx = torch.tensor([f.sub_idx for f in train_features], dtype=torch.long)
all_obj_idx = torch.tensor([f.obj_idx for f in train_features], dtype=torch.long)
train_data = TensorDataset(all_input_ids, all_input_mask, all_segment_ids, all_label_ids, all_sub_idx, all_obj_idx)
train_dataloader = DataLoader(train_data, batch_size=train_batch_size)
train_batches = [batch for batch in train_dataloader]

num_train_optimization_steps = len(train_dataloader) * num_train_epochs

logger.info("***** Training *****")
logger.info("  Num examples = %d", len(train_examples))
logger.info("  Batch size = %d", train_batch_size)
logger.info("  Num steps = %d", num_train_optimization_steps)

best_result = None
eval_step = max(1, len(train_batches) // eval_per_epoch)

lr = learning_rate
model = RelationModel.from_pretrained(
    'allenai/scibert_scivocab_uncased', cache_dir=str(
    PYTORCH_PRETRAINED_BERT_CACHE), num_rel_labels=num_labels)
if hasattr(model, 'bert'):
    model.bert.resize_token_embeddings(len(tokenizer))
elif hasattr(model, 'albert'):
    model.albert.resize_token_embeddings(len(tokenizer))
else:
    raise TypeError("Unknown model class")

model.to(device)
if n_gpu > 1:
    model = torch.nn.DataParallel(model)

param_optimizer = list(model.named_parameters())
```

```
no_decay = ['bias', 'LayerNorm.bias', 'LayerNorm.weight']
optimizer_grouped_parameters = [
    {'params': [p for n, p in param_optimizer
                if not any(nd in n for nd in no_decay)], '
    weight_decay': 0.01},
    {'params': [p for n, p in param_optimizer
                if any(nd in n for nd in no_decay)], 'weight_decay':
    0.0}
]
optimizer = AdamW(optimizer_grouped_parameters, lr=lr, correct_bias=
    not(bertadam))
scheduler = get_linear_schedule_with_warmup(optimizer, int(
    num_train_optimization_steps * warmup_proportion),
    num_train_optimization_steps)

start_time = time.time()
global_step = 0
tr_loss = 0
nb_tr_examples = 0
nb_tr_steps = 0
for epoch in range(int(num_train_epochs)):
    model.train()
    logger.info("Start epoch #{} (lr = {})...".format(epoch, lr))
    if train_mode == 'random' or train_mode == 'random_sorted':
        random.shuffle(train_batches)
    for step, batch in enumerate(train_batches):
        batch = tuple(t.to(device) for t in batch)
        input_ids, input_mask, segment_ids, label_ids, sub_idx,
    obj_idx = batch
        loss = model(input_ids, segment_ids, input_mask, label_ids,
    sub_idx, obj_idx)
        if n_gpu > 1:
            loss = loss.mean()

        loss.backward()

        tr_loss += loss.item()
        nb_tr_examples += input_ids.size(0)
        nb_tr_steps += 1

        optimizer.step()
        scheduler.step()
        optimizer.zero_grad()
```

```
        global_step += 1

        if (step + 1) % eval_step == 0:
            logger.info('Epoch: {}, Step: {} / {}, used_time = {:.2f}
    s, loss = {:.6f}'.format(
                        epoch, step + 1, len(train_batches),
                        time.time() - start_time, tr_loss /
    nb_tr_steps))
            save_model = False
            if do_eval:
                preds, result, logits = evaluate(model, device,
    eval_dataloader, eval_label_ids, num_labels, e2e_ngold=eval_nrel)
                model.train()
                result['global_step'] = global_step
                result['epoch'] = epoch
                result['learning_rate'] = lr
                result['batch_size'] = train_batch_size

                if (best_result is None) or (result[eval_metric] >
    best_result[eval_metric]):
                    best_result = result
                    logger.info("!!! Best dev %s (lr=%s, epoch=%d):
    %.2f" %
                                (eval_metric, str(lr), epoch, result[
    eval_metric] * 100.0))
                    save_trained_model(output_dir, model, tokenizer)
```

Code 3: Training the relation model from scratch

```
def compute_f1(preds, labels, e2e_ngold):
    n_gold = n_pred = n_correct = 0
    for pred, label in zip(preds, labels):
        if pred != 0:
            n_pred += 1
        if label != 0:
            n_gold += 1
        if (pred != 0) and (label != 0) and (pred == label):
            n_correct += 1
    if n_correct == 0:
        return {'precision': 0.0, 'recall': 0.0, 'f1': 0.0}
    else:
        prec = n_correct * 1.0 / n_pred
        recall = n_correct * 1.0 / n_gold
        if prec + recall > 0:
```

```
            f1 = 2.0 * prec * recall / (prec + recall)
        else:
            f1 = 0.0

        if e2e_ngold is not None:
            e2e_recall = n_correct * 1.0 / e2e_ngold
            e2e_f1 = 2.0 * prec * e2e_recall / (prec + e2e_recall)
        else:
            e2e_recall = e2e_f1 = 0.0
        return {'precision': prec, 'recall': e2e_recall, 'f1': e2e_f1
    , 'task_recall': recall, 'task_f1': f1,
        'n_correct': n_correct, 'n_pred': n_pred, 'n_gold': e2e_ngold
    , 'task_ngold': n_gold}
```

Code 4: Helper function to compute F1 scores