INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO
MESTRADO EM ENGENHARIA DE INTELIGÊNCIA ARTIFICIAL




NERdy: Melhoria da Descoberta de Informação através do Reconhecimento de Entidades

JOÃO VILAS BOAS DA SILVA MAGALHÃES
Setembro de 2024

P.PORTO

NERdy

Enhancing Information Discovery through Named Entity Recognition

João Vilas Boas da Silva Magalhães

Student no: 1181053

Dissertation for the degree of Master of Artificial Intelligence Engineering

Supervisor:

Luiz Felipe Rocha de Faria, Professor Coordenador do Instituto Superior de Engenharia do Porto do Instituto Politécnico do Porto

Júri:

Presidente:

Joaquim Filipe Peixoto dos Santos, Professor Adjunto do Instituto Superior de Engenharia do Porto do Instituto Politécnico do Porto

Vogais:

Luís Manuel Silva Conceição, Professor Adjunto do Instituto Superior de Engenharia do Porto do Instituto Politécnico do Porto

Luiz Felipe Rocha de Faria, Professor Coordenador do Instituto Superior de Engenharia do Porto do Instituto Politécnico do Porto

Porto, September 2024

"*In the darkest times, hope is something you give yourself.*
That is the meaning of inner strength."

Iroh, Book Two: Earth (土)

Abstract

Education is essential for individual and societal progress, playing a pivotal role in economic development, creativity, and social mobility. However, a significant challenge remains in ensuring equitable access to quality education, particularly in diverse classrooms where personalized learning is increasingly critical. Research highlights the benefits of tailored learning approaches, but current educational tools often lack the ability to organize raw information into structured formats suitable for individualized learning. This gap underscores the need for advancements in Natural Language Processing (NLP) to enhance educational tools.

In response to this challenge, this project focuses on developing a Named Entity Recognition (NER) model to improve the organization and extraction of information from raw text. NER, a key task in NLP, identifies and classifies entities such as people, organizations, and locations, providing the groundwork for future tools designed to structure educational content. The primary objective of this study is to construct an entity extraction tool, with the ultimate goal of enhancing personalized learning by facilitating the automatic organization of educational materials.

To achieve this, a model combining a pretrained BERT encoder, a BiLSTM layer, and a Conditional Random Field (CRF) correction layer was developed. The model was trained on curated datasets to ensure both performance and fairness. Through extensive testing and fine-tuning, the model demonstrated strong results, achieving an F1 score of 87.22%, comparing favorably to state-of-the-art models. Key techniques such as class balancing, weight decay, and dropout were used to prevent overfitting, while validation and training losses were monitored to assess the model's performance.

The findings of this project not only confirm the effectiveness of the developed NER model but also highlight its potential in addressing educational challenges. The model shows promise for future expansion, including the development of relation extraction techniques and knowledge graph generation to further enhance learning tools. Ethical considerations, including data privacy, fairness, and transparency, were prioritized throughout the project. Future work will focus on refining the model and expanding its capabilities to better serve the educational sector, contributing to the broader goal of improving access to quality, personalized education.

Keywords: Artificial Intelligence, BERT, BiLSTM, CRF, Learning Styles, Named Entity Recognition, Natural Language Processing, Personalized Learning, Relation Extraction

Resumo

A educação é essencial para o progresso individual e societal, desempenhando um papel fundamental no desenvolvimento económico, na criatividade e na mobilidade social. No entanto, permanece um desafio significativo em garantir o acesso equitativo a uma educação de qualidade, particularmente em salas de aula diversificadas, onde a aprendizagem personalizada é cada vez mais crítica. A investigação destaca os benefícios das abordagens de aprendizagem personalizadas, mas as ferramentas educativas atuais muitas vezes carecem da capacidade de organizar informação bruta em formatos estruturados adequados para a aprendizagem individualizada. Esta lacuna sublinha a necessidade de avanços no Processamento de Linguagem Natural (NLP) para melhorar as ferramentas educacionais.

Em resposta a este desafio, este projeto foca-se no desenvolvimento de um modelo de Reconhecimento de Entidades Nomeadas (NER) para melhorar a organização e extração de informação de texto bruto. O NER, uma tarefa chave no NLP, identifica e classifica entidades como pessoas, organizações e localizações, proporcionando a base para ferramentas futuras destinadas à estruturação de conteúdo educacional. O principal objetivo deste estudo é construir uma ferramenta de extração de entidades, com o objetivo final de melhorar a aprendizagem personalizada através da facilitação da organização automática de materiais educacionais.

Para alcançar este objetivo, foi desenvolvido um modelo que combina um codificador BERT pré-treinado, uma camada BiLSTM e uma camada de correção com Campos Aleatórios Condicionais (CRF). O modelo foi treinado em conjuntos de dados selecionados para garantir tanto o desempenho quanto a equidade. Através de testes extensivos e ajustes finos, o modelo demonstrou resultados sólidos, alcançando uma pontuação F1 de 87,22%, comparando-se favoravelmente com modelos de ponta. Técnicas chave, como balanceamento de classes, decaimento de peso e dropout, foram utilizadas para prevenir overfitting, enquanto as perdas de validação e treino foram monitorizadas para avaliar o desempenho do modelo.

Os resultados deste projeto não só confirmam a eficácia do modelo NER desenvolvido, como também destacam o seu potencial para enfrentar desafios educacionais. O modelo mostra-se promissor para futuras expansões, incluindo o desenvolvimento de técnicas de extração de relações e a geração de grafos de conhecimento para melhorar ainda mais as ferramentas de aprendizagem. Considerações éticas, incluindo privacidade de dados, equidade e transparência, foram priorizadas ao longo do projeto. O trabalho futuro centrar-se-á em refinar o modelo e expandir as suas capacidades para melhor servir o setor educacional, contribuindo para o objetivo mais amplo de melhorar o acesso a uma educação personalizada e de qualidade.

Palavras-chave: Artificial Intelligence, BERT, BiLSTM, CRF, Learning Styles, Named Entity Recognition, Natural Language Processing, Personalized Learning, Relation Extraction

Acknowledgements

This project marks not only the end of a year of work and stress, but the end of an era—my academic era. One that has spanned two decades, my entire life, in fact. While I will never stop learning, changing, and growing as a person, I understand that the time for rebirth is closer than ever. For it is not courageous to feel no fear as you face life, but to feel fear and face it anyway. When nothing is true and everything is permitted, all you can do is take a leap of faith.

ISEP has proven to be a second home, but it is with pain that I recall saying goodbye to my second family three years ago as they left to chase life. It is a different pain I feel this year, as I too leave this home, the memories still attached. Thank you, ISEP.

To my family, who has supported me selflessly all my life, giving me the best opportunity to reach my potential—though I didn't always do my best, I have lived life in earnest, and I will continue to grow and develop to the best of my understanding. I hope I never have to say goodbye to you. Thank you, family.

To my friends, who have accompanied me through the best and the worst, reminding me of my strength when I felt weakest—I hope to give back to you what you have given to me. Thank you, friends.

To my supervisor, for supporting me through this project and helping me complete this chapter of my life. I hope to someday accomplish the project I envisioned. Thank you, professor.

Finally, to myself, for not giving up in the darkest hour. For understanding that life is made of perspectives and that people can change. For starting to take control and beginning to believe. You can be the master of your fate and the captain of your soul. But you must realize that life comes from you, not at you. I hope one day to fully understand these words. Thank you.

I hope…

Thank you.

Table of Contents

List of Figures

List of Tables

Acronyms and Symbols

Acronym	Definition
ACE	Automated Concatenation of Embeddings
AI	Artificial Intelligence
API	Application Programming Interface
BERT	Bidirectional Encoder Representations from Transformers
BGRU	Bidirectional Gated Recurrent Unit
BiLSTM	Bidirectional Long Short-Term Memory
CNN	Convolutional Neural Network
CR-CNN	Classification by Ranking Convolutional Neural Network
CRF	Conditional Random Field
CRISP-DM	Cross Industry Standard Process for Data Mining
DL	Deep Learning
DQN	Deep Q Network
ERNIE	Enhanced Representation through Knowledge Integration
FCM	Factor-based Compositional Embedding Model
FD	Field-Dependent
FI	Field-Independent
FLERT	Feature-Level Representations for Named Entity Recognition
FN	False Negative
FP	False Positive
GCN	Graph Convolutional Networks
GRU	Gated Recurrent Unit
HMM	Hidden Markov Model
IDCNN	Impulse Detection Convolutional Neural Network
IDE	Integrated Development Environment
KG	Knowledge Graph
KRL	Knowledge Representation Learning
LSTM	Long Short-Term Memory
ML	Machine Learning
MLM	Masked Language Model
MVRNN	Matrix-Vector Recursive Neural Network
NaN	Not a Number
NED	Named Entity Disambiguation
NEL	Named Entity Linking
NER	Named Entity Recognition
NLP	Natural Language Processing
POS	Part-of-Speech
PRISMA	Preferred Reporting Items for Systematic Reviews and Meta-Analyses
RE	Relation Extraction
RNN	Recurrent Neural Network
SVM	Support Vector Machines
TN	True Negative
TP	True Positive

Chapter 1

Introduction

This initial chapter provides an introduction to the topic subject of this project. It contextualizes the reader and introduces the problem statement, which this project will address and attempt to provide a solution for. It also lists the project's objectives according to the given problem and, finally, provides a structure for the whole document as a final section to the chapter.

1.1 Contextualization

Education is a cornerstone of development, essential for sustainable economic progress and individual well-being. It enriches understanding, boosts productivity, fosters creativity, and plays a vital role in securing economic and social advancements, ensuring a balanced and prosperous society [1], [2]. Education, particularly higher education such as university degrees, significantly influences income growth and economic development [3].

Several studies highlight the crucial role of personalized learning and teaching [4], [5], [6], [7], [8]. These findings emphasize the inherent diversity among students, indicating that varied learning strategies yield better results for different individuals. The experiment in [4] examines how intuitive and analytical learners experience varied outcomes using a chatbot. The study reveals differences in learning achievements, motivation, collective efficacy, cognitive and emotional engagement, and satisfaction with the learning approach between the two groups. By analyzing Moodle logs, [5] identifies four distinct patterns among students based on their interaction with quizzes and clickstreams on lesson slide notes. The experiment in [6] explores the distinction between field-dependent and field-independent students and evaluates the effectiveness of teaching approaches tailored to each group, specifically comparing active and passive learning methods.

Natural Language Processing (NLP) is a field of artificial intelligence (AI) that focuses on enabling computers to understand, interpret, and generate human language. In the realm of NLP, two

essential tasks are named entity recognition (NER) and relation extraction (RE) [9], [10], [11], [12].

NER involves identifying and categorizing specific entities or objects mentioned in a given text [9], [13], [14], [15], [16], [17], [18], [19]. These entities can range from people and places to organizations, dates, and more. By extracting entities, NLP systems can create a structured representation of the information contained in a text, facilitating further analysis, and understanding, making NER a key task within these systems.

RE, on the other hand, delves into establishing connections or associations between different entities identified in a text [9], [10], [11], [12], [20], [21], [22], [23]. This task aims to determine the nature of the relationship between pairs of entities, providing valuable insights into the contextual meaning of the information.

Knowledge graphs (KG) also play a crucial role in organizing and representing the extracted entities and their relationships identified through NER and RE. A KG is a structured knowledge base that uses graph structures to depict entities as nodes and their relationships as edges. It provides a semantic framework for connecting disparate pieces of information, making it easier to navigate and comprehend complex datasets [9], [24].

In practical applications, combining NER, RE, and KG enhances various processes such as information retrieval, question answering, and automated decision-making. For example, in a medical context, NLP can extract entities like symptoms and diseases from clinical notes, classify relations between them, and construct a KG that aids healthcare professionals in diagnosing and treating patients.

Overall, the synergy between NLP, NER, RE, and KG empowers machines to better understand and interpret human language, opening avenues for more sophisticated and context-aware applications across diverse domains. While RE and KG play important roles, this study primarily focuses on advancing NER as a foundational step towards further development in educational and organizational tools.

1.2 Problem

A significant global challenge persists with a considerable number of people lacking education [2], stemming from various factors such as insufficient access to proper educational resources, disinterest in pursuing education, or individuals dropping out of school prematurely. Despite efforts to enhance worldwide education, a substantial portion of the population remains uneducated, hindering personal development and limiting opportunities for social and economic advancement [2].

Ensuring proper learning support for students to organize knowledge is crucial, and the integration of AI, despite its limitations, can enhance this process. The exploration of leveraging messages through NLP techniques offers potential improvements in educational contexts [4]. This relates directly to NER techniques, a crucial aspect of NLP, whose relevance is highlighted when considering educational issues, mainly those regarding to information extraction and organization. Additionally, the acknowledgment of diverse student profiles in [6] emphasizes the importance of personalized learning techniques. Recognizing individual differences and tailoring approaches accordingly can contribute to more effective and inclusive educational

experiences. On the other hand, personalized teaching can contribute to disparities within the classroom, favoring certain students over others. This unequal treatment may result in the marginalization of individuals who require different learning approaches [2].

Integrating technology into teaching requires significant upfront effort to familiarize teachers with software, instructional strategies, and group facilitation using technology. Many educators, especially those with traditional backgrounds, face a steep learning curve in adapting to tech-based instruction [2]. The challenges extend to a lack of comprehensive professional development, hindering their ability to teach in a personalized manner without adequate technology support [25].

The research conducted for this document reveals a scarcity of applications that effectively take raw information and organize it into structured formats suitable for learning and education, especially in the context of personalized learning and easily understandable materials. NER techniques play a critical role in addressing this need by creating the steppingstone for organizing and structuring educational content. By focusing on these advancements, this study contributes towards developing solutions that can eventually organize and structure educational content. While similar applications will be presented later in the report, it's crucial to note that they may not align with the specific purpose envisioned for the application discussed here. In studying these applications, we understand the importance of NER as a foundational step in achieving more sophisticated educational tools.

1.3 Objectives

The main objective of this thesis is the following:

- **MO:** Construct an entity extraction tool.

This tool should use NER to process raw text input, extracting key entities from the text, serving as a foundation for future information structuring tools. To support this main objective, the following secondary objectives have been defined:

- **SO1:** Investigate and identify the most effective techniques for building an NER model tailored to the project's requirements.
- **SO2:** Implement an NER model to identify and classify entities in various text inputs.
- **SO3:** Evaluate and fine-tune the NER mechanism to improve its accuracy and adaptability across different datasets.

To further detail the project's focus, the following research questions were formulated to guide both the research and implementation processes, as seen in Table 1. These questions ensure that the project remains aligned with its objectives and addresses the key areas of interest necessary for the successful development of the entity extraction tool. While these questions will be further explored in the State-of-the-Art chapter, where the research will be discussed in more depth, they are introduced here because they also directly pertain to the objectives of the project.

Table 1 Research Questions

ID	Question
RQ1	What are the most effective methods and models in the implementation of NER in modern NLP systems?
RQ2	What are the most compatible RE methods for integrating with NER?
RQ3	How does NER relate to different learning styles in educational settings?
RQ4	What is the most effective approach for presenting information to accommodate diverse learning styles?

1.4 Document Structure

This document is organized into six chapters: Introduction, State of the Art, Methodology, BERT-BiLSTM-CRF, Results and Conclusion. The introductory chapter provides an overview of the topic and outlines high-level objectives. The State of the Art chapter delves into current and relevant technologies, presenting findings that contribute to the project and its implementation. The Methodology chapter details the methods employed in developing the project and its implementation. BERT-BiLSTM-CRF showcases the project's model design process with the analyses for its metrics being explored in the Results chapter. Finally, the Conclusion chapter gives a summary of the findings of the paper as well as addresses future work.

Chapter 2

State of the Art

The upcoming chapter provides an overview of the project's state of the art, employing a systematic review approach guided by the Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) method. This review is aligned with the project’s objectives, focusing on the research of information useful to its implementation.

The chapter commences by detailing the research methods employed in the systematic review, followed by the presentation of results derived from the examination of relevant articles, concluding with a section discussing and exploring these findings.

2.1 Methods

This section initiates by exploring the pivotal research questions that will guide this project’s implementation strategy. The formulation of these questions plays a crucial role in shaping the project's approach. Subsequently, we provide a detailed specification of the data sources instrumental in collecting relevant articles. This is followed by a listing of the search terms employed during the retrieval process and the creation of specific query strings from these terms. Moving forward, the Quality Assessment section outlines the criteria employed to make informed decisions regarding the inclusion or exclusion of articles. These criteria serve as a methodological framework, ensuring a systematic and rigorous approach to selecting articles that align with the project's objectives.

2.1.1 Research Questions

This review aligns seamlessly with the research problem under investigation, prompting the formulation of four succinct questions to refine and streamline the research process, as seen in Table 2.

The first question focuses on identifying the most effective methods and models for NER, which is critical to the project's primary objective of extracting key entities from raw text.

The second question investigates the potential compatibility of RE methods with NER, examining how these two technologies can work together to expand the functionality of the tool.

Shifting focus, the third question explores how different NER methods might relate to various learning styles among students.

Lastly, the fourth question seeks to identify the optimal approach for displaying information that caters to diverse learning styles.

Table 2 Research Questions

ID	Question
RQ1	What are the most effective methods and models in the implementation of NER in modern NLP systems?
RQ2	What are the most compatible RE methods for integrating with NER?
RQ3	How does NER relate to different learning styles in educational settings?
RQ4	What is the most effective approach for presenting information to accommodate diverse learning styles?

2.1.2 Data Sources

To address the research questions, an analysis of various articles was conducted, drawing from specific data sources outlined in Table 3. IEEE Xplore was chosen for technological research due to its strong reputation in engineering. ScienceDirect was utilized for social research, specifically focusing on learning styles. Additionally, Google Scholar served as a comprehensive platform for any supplementary manual searches conducted during the research process.

Table 3 Data sources

Database	URL	Usage
IEEE Xplore	https://ieeexplore.ieee.org	NLP
ScienceDirect	https://www.sciencedirect.com	Learning Styles
Google Scholar	https://scholar.google.pt	Manual searches

2.1.3 Search Terms

In configuring search strategies for each data source, specific search terms were chosen to initiate the data retrieval process, as seen in Table 4. The selected terms were intentionally broad to attract a diverse range of articles, with the intention of later refining the search results to eliminate irrelevant content during the analysis phase.

Table 4 Search Terms

ID	Data Source	Keywords
D1	IEEE Xplore	Entity Extraction, NLP
D2	ScienceDirect	Learning, Methods; Styles

Each data source presented different capabilities in terms of search options, leading to distinct query structures, displayed in Table 5. For IEEE Xplore, the search focused on ensuring "entity extraction" appeared in the abstract, considering its critical relevance to the research, while "NLP" could appear anywhere in the metadata of the articles. In contrast, the search on ScienceDirect was more general, given the platform's more rigid search options. It targeted the title, abstract, or keywords for terms related to learning methods or learning styles, providing flexibility in gathering relevant articles for the social aspect of the research.

Table 5 Query Strings

ID	Query String
D1	("Abstract":entity extraction) AND ("All Metadata":nlp)
D2	Title, abstract, keywords: learning AND (methods OR styles)

2.1.4 Quality Assessment

In the process of screening and selecting articles for the analysis, a set of inclusion and exclusion criteria was employed to determine whether an article would be removed or not. The specific

inclusion criteria applied are detailed in Table 5, while the exclusion criteria can be found in Table 7.

Table 6 Inclusion Criteria

ID	Criteria
IC1	The source focuses on education
IC2	The source focuses on thinking skills
IC3	The source focuses on teaching and teacher education
IC4	The source focuses on social skills
IC6	The source focuses on NER
IC7	The source focuses on RE
IC8	The source focuses on NLP

Table 7 Exclusion Criteria

ID	Criteria
EC1	The source is older than 6 years (published before 2018)
EC2	The source is not written in English
EC3	The source is not peer-reviewed
EC4	The source is a pre-proof work
EC5	The source does not focus on the technology it uses

2.1.5 Data Extraction

With the defined search parameters, the search was initiated, employing the appropriate query string for each data source while applying date and language exclusion criteria in the search itself. The number of results is outlined in Table 8. From each source, the most pertinent 100 results were selected for the screening phase. No duplicates were found since the two searches varied in topic drastically.

Table 8 Search results

Data source	No. results
IEEE Xplore	181
ScienceDirect	408
Total	589

It is important to mention that manually picked articles, retrieved mainly from Google Scholar skip the entire screening process. These articles followed the same exclusion and inclusion criteria but were not included in the selection process.

For the screening process, the Rayyan software was utilized, providing a platform for efficiently managing the list of results. This tool enabled users to swiftly visualize the title and abstract of each entry, offering options to include or exclude entries. Additionally, it facilitated the application of relevant tags to entries, enabling a more organized view. To commence, each entry was tagged with descriptors such as the article's topic, utilized technology, length, and perceived relevance. Following the tagging process, the screening was initiated, excluding articles aligning with the exclusion criteria and including those meeting the inclusion criteria. An overview of this process is illustrated in the diagram presented in Figure 1, following the flowchart methodology outlined in PRISMA.

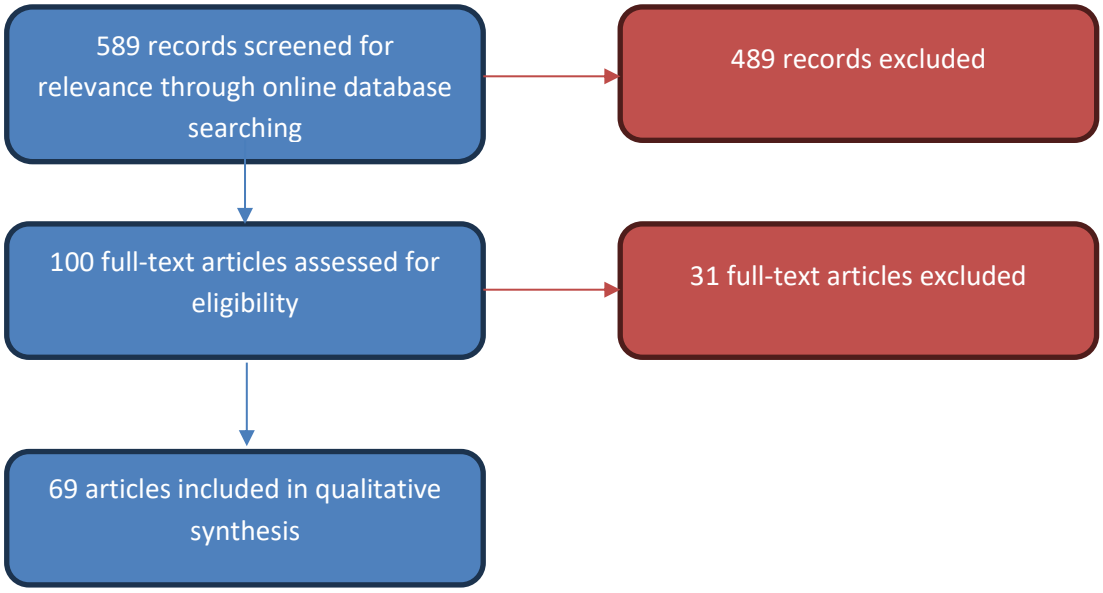

Figure 1 - PRISMA flowchart

2.2 Results

This section presents the findings obtained through the examination of the articles, addressing each research question posed.

2.2.1 What are the most effective methods and models in the implementation of NER in modern NLP systems?

Many of the examined articles discuss distinct approaches for implementing NER: a rule-based approach, a machine learning (ML)-based approach and a deep learning (DL) approach, though most of the articles that mention ML approaches use them in tandem with DL approaches [15], [17], [19], [26].

The rule-based approach to NER primarily relies on the formulation of manually crafted rules by a human expert [15]. An inherent advantage of such methods lies in their independence from annotated training data, leveraging instead the richness of lexical resources [15]. Furthermore, the precision of handcrafted methods tends to be high, owing to the incorporation of lexicons and domain-specific knowledge [15]. However, these merits come with certain drawbacks. Notably, rule-based approaches exhibit a degree of domain dependency [15], [19], and any alterations to the rules necessitate corresponding adjustments to the algorithm [19]. Additionally, the process of creating intricate rules can be time-consuming, representing a notable challenge associated with this approach.

The ML-and DL-based approaches to NER endow computers with the capacity to learn autonomously [13], [19]. These algorithms undergo training on extensive datasets, enabling them to formulate predictions for new data [14]. The ML approach excels in scalability, accommodating diverse languages [19], surpasses the accuracy of rule-based methods [13], requires minimal human intervention in training [13], and offers cost-effective maintenance compared to rule-based systems [19]. However, the ML approach may incur higher computational expenses [13], [19], and its accuracy is contingent on the comprehensiveness of the training data, potentially yielding lower precision if the dataset lacks thorough representation [13].

Rule-based approaches for NER can be further categorized into dictionary-based, pattern-based, and grammar-based methods. Dictionary-based NER utilizes a predefined dictionary to identify entities in text [27], which can be manually created or automatically generated from a corpus of text [28]. It ensures high precision with handcrafted rules [27], [28], [29] and is efficient for small datasets [15], [29]. However, it shows difficulty in scaling to large datasets [15], [29] and is domain dependent [15], [28]. Pattern-based NER employs regular expressions, such as Regex, to identify entities in text by recognizing specific patterns [19]. This approach is akin to dictionary-based extraction but is less labor-intensive and more versatile, relying on patterns rather than specific words for identification [19]. Finally, the grammar-based approach is even broader than the pattern-based method, as it encompasses the rules for an entire language; however, it is language-dependent [19], [27].

Within ML and DL approaches, several models are used and tested for quality comparison among the various articles.

Two models are utilized in [30]: a convolutional neural network (CNN)- Bidirectional Encoder Representations from Transformers (BERT) model and a simple BERT model, working together to extract features from the input text and identify entities. In [13], the pretraining layer is handled by BERT, followed by a Bidirectional Long Short-Term Memory (BiLSTM) layer and an Impulse Detection Convolutional Neural Network (IDCNN) layer for feature extraction. The resulting outputs are merged and input into a Conditional Random Field (CRF) layer for correction. For NER on Bangla online newspapers, [17] employs a Gated Recurrent Unit (GRU) within a Recurrent Neural Network (RNN) model but achieves less impressive results. Highlighting the importance of linking NER with Named Entity Disambiguation (NED) and Named Entity Linking (NEL), [15] points out a lack of universally accepted evaluation methods and challenges in comparing pure NER with combined approaches. Many NER systems are tailored for a limited set of entity types, posing challenges for reconfigurability. Transfer-learning approaches are proposed as a promising avenue for future research to address this limitation. In the comparison of various neural models for NER in [18], including CNN-LSTM, BiLSTM, and BiLSTM-CRF, alongside pre-trained language models like word2vec and BERT, results indicate that the BERT-BiLSTM-CRF model achieved the highest performance, boasting an F1 score of approximately 75%. The authors in [19] employs a Hidden Markov Model (HMM) combined with a rule-based approach for NER, demonstrating the effectiveness of rule-based methods, albeit with slightly lower efficiency. Lastly, [31] leverages BERT, Feature-Level Representations for Named Entity Recognition (FLERT), Automated Concatenation of Embeddings (ACE), LSTM with CRF for NER.

2.2.2 What are the most compatible RE methods for integrating with NER?

RE is intricately linked with NER, and various models have been explored in projects focusing on both aspects. This overview delves into several models and methods discussed in research articles, particularly emphasizing their compatibility with NER.

In [10], the author advocates for dependency-based approaches, comparing sequence-based, tree-based, and graph models. Sequence-based models excel at capturing sequential dependencies but may struggle with long-range dependencies. Tree models preserve syntactic structures but can be computationally inefficient for large trees. Graph models, however, offer flexibility, efficiently encoding both sequential and hierarchical dependencies, making them suitable with NER systems. The proposed model in [10] integrates a BiLSTM [21] network with a dependency propagation layer using parallel matrices, culminating in a feed-forward neural network for RE.

The author in [32] critiques traditional ML approaches for RE and emphasizes the superiority of DL methods such as Matrix-Vector Recursive Neural Network (MVRNN), CNN, Factor-based Compositional Embedding Model (FCM), and Classification by Ranking Convolutional Neural Network (CR-CNN). Additionally, pre-training models like BERT and Enhanced Representation through Knowledge Integration (ERNIE) are introduced to leverage semantic knowledge from unlabeled text, enhancing their effectiveness in conjunction with NER.

This method of pre-training models with BERT is used in several other articles [12], [20], [32], [33]. In [20], BERT is utilized for pre-training data using a Masked Language Model (MLM). The model replaces BiLSTM with a transformer encoder, employs Bidirectional Gated Recurrent Unit (BGRU) for contextual information, and integrates part-of-speech and keyword

information in the attention network. A CRF layer is added for label constraints, reinforcing the model's integration potential with NER systems.

Authors in [12] express concerns that existing methods often overlook entity information and domain knowledge, particularly the directional aspect of entity relationships. They propose BERT-KRL, integrating Knowledge Representation Learning (KRL) to efficiently calculate semantic connections between entities in low-dimensional space. The model achieves state-of-the-art results, especially surpassing models using Graph Convolutional Networks (GCN) and LSTM for RE, indicating its compatibility with NER frameworks.

Addressing challenges in RE, [34] introduces the Distant Supervision strategy, assuming that sentences mentioning entities with a known relation in knowledge bases express that relation. Multi-head self-attention is employed to capture long dependencies, and curriculum learning with a label denoising mechanism enhances precision. The Latent-label denoising method corrects noisy labels, proving effective across different representation learning networks, which could complement NER tasks effectively.

Similarly, [22] acknowledges challenges with distant supervision, highlighting the problem of wrong labeling. They introduce a label denoiser using a Deep Q Network (DQN) to select reliable labels, achieving noise reduction and outperforming baselines on multiple datasets, further indicating compatibility with NER applications.

Finally, [23] emphasizes the core role of BiLSTM networks, addressing the vanishing gradient problem and introducing a word-level attention mechanism. An entity tensor layer, employing a bilinear form, is integrated to capture deeper interactions between entities, showcasing its relevance for enhancing NER outcomes.

2.2.3 How does NER relate to different learning styles in educational settings?

Several of the reviewed articles consistently advocate the notion that students exhibit distinct cognitive styles [4], [5], [6], [7], [35], [36], [37], [38]. Moreover, there is a prevailing consensus among these articles that aligning teaching strategies with individual student cognitive styles yields optimal learning outcomes [6], [7], [8], [36], [39], [40].

The author in [4] distinguishes between learning style and cognitive style. While learning style relates to one's preferred approach to learning activities, cognitive style involves an individual's stable method of decision-making and mental information processing [4], [6]. In the context of NER, understanding these styles can help tailor instructional methods. For example, analytical learners, who process information linearly through scanning and critiquing may benefit from structured tutorials and systematic coding exercises in NER applications, whereas intuitive learners who prefer exploratory projects that allow them to discover ideas holistically from new or scattered information may align more closely with the methodologies used in modern NER systems. These systems leverage deep learning and exploratory data analysis, requiring pattern recognition and an understanding of complex datasets, which resonates with the intuitive learner's approach.

Another author distinguishes between two cognitive styles, categorizing individuals as either field-dependent (FD) or field-independent (FI) [6]. FD individuals demonstrate sensitivity to environmental cues, interpreting information within the given context. On the other hand, FI

individuals rely on internal references, extracting information irrespective of its background context. In comparison, FI individuals exhibit heightened autonomy, freedom, and enhanced analytical skills when approaching problems. While there isn't a direct correspondence, parallels can be drawn between FI individuals and analytical learners, as both share similarities in their preference for analytical reasoning. Similarly, FD individuals align with intuitive learners, as they both tend to rely on holistic perceptions influenced by environmental cues [4], [6]. In regard to NER techniques, FD individuals may thrive in contexts where the model leverages contextual information from the data, such as through attention mechanisms, while FI individuals might excel with methods that emphasize independent analysis, like feature extraction and pattern recognition.

A more dynamic perspective is adopted in [5] on learners by classifying them into four patterns based on two distinct metrics: the amount of study and general test scores. Instead of focusing on specific learning styles, the paper offers a flexible approach to supervising students and assessing their success in their study. This flexible approach can help in adapting NER techniques to suit different student needs, acknowledging variability in their learning process.

The most clearly defined list of learning styles was identified in [37]. It included independent, social, audio-visual, active, verbal, logical, and intuitive. Recognizing these styles can enhance the application of NER in teaching by ensuring an appropriate and extensive environment adapted to different needs. The article highlights that barriers to critical thinking can have adverse effects on both self-leadership and electronic learning styles. To improve overall efficiency, the study suggests the incorporation of concepts such as self-leadership and appropriate learning styles.

2.2.4 What is the most effective approach for presenting information to accommodate diverse learning styles?

Many articles emphasize the implementation of techniques designed to accommodate various types of students, aiming to engage and cater to diverse learning preferences [4], [7], [8], [36], [39], [40]. While acknowledging that these techniques may not be equally effective for every student, the overarching goal is to enhance learning possibilities for the entire student population. In the context of NER, customizing tools of applications to adapt to these diverse learning styles could support effective and personalized learning experiences.

The study in [40] explored the effectiveness of adopting a conversational teaching approach, aiming to create a less formal and more fluid learning environment. The findings indicated that students exhibited increased attentiveness, engagement, and motivation within this approach. In NER-based applications, conversational interfaces (such as chatbots) could be designed to dynamically adjust their complexity and pace based on user interactions, helping to accommodate students who benefit from a more engaging and informal structure. However, the study also revealed that the faster paced setting induced higher levels of stress and pressure among students. This implies that an NER system should be mindful of cognitive overload with adaptive pacing mechanisms, where a system would recognize when a student needed to slow down.

Similarly, in [4] a conversational learning approach was implemented facilitated by a chatbot. Notably, it revealed distinctions in academic achievements based on different cognitive styles,

but also, and more importantly showcased a marked contrast between students utilizing the chatbot and those following a traditional approach. The chatbot-engaged students demonstrated superior results compared to their counterparts in the conventional learning setting. Furthermore, resembling [40], the study indicated that highly engaged students experienced a heightened mental load, emphasizing a connection between increased engagement and cognitive demands.

Both studies highlight how highly engaged students often experience heightened mental load. In NER systems, this suggests that balancing engagement with cognitive capacity demands is crucial. Effective use of NER would involve incorporating user adaptability features which adjust the system based on how much information the student is ready to process at a given time.

2.3 Discussion

The following section serves as a summary for the results gathered from the examined articles, delving into specific topics relevant to the current project.

2.3.1 NLP

A future use of the work done in this project is to create a KG. A KG represents data as triplets composed as (s, p, o): subject s, predicate p, object o [15]. This framework allows displays of information in a multitude of ways. By processing natural language input, NLP techniques such as pre-processing, NER and RE play an essential role in preparing data for potential future KG construction.

Pre-Processing

Pre-processing in NLP encompasses various techniques aimed at refining raw text data before it undergoes further analysis. Commonly employed pre-processing methods include:

- Tokenization: Breaking down text into individual tokens or words [41], [42], [43], [44].
- Part-of-Speech (POS) Tagging: Assigning grammatical categories to each token, such as noun, verb, adjective, etc. [41], [42], [44], [45], [46], [47].
- Stop-word Removal: Eliminating common and non-informative words that do not contribute significantly to the analysis [48].
- Normalization: Standardizing text by converting it to lowercase, for uniformity [42].
- Sentence Splitting: Dividing the text into individual sentences [42], [43], [47].
- Lemmatization: Reducing words to their base or root form to enhance consistency [41].
- Chunking and Dependency Parsing: Identifying and grouping words based on their syntactic relationships [47].
- Structural Parsing: Analyzing the grammatical structure of sentences [42].

Some research studies may not explicitly mention the pre-processing techniques utilized. This could be attributed to the use of pre-processed datasets or a reliance on DL methods within NLP [15]. DL approaches often demonstrate a reduced reliance on extensive pre-processing due to their ability to automatically learn intricate patterns from raw data, mitigating the need for manual intervention.

NLP methods

When employing NLP in an application, three different types are considered, each with different characteristics and considerations.

Rules-based approach relies on predefined rules and lexical resources. While achieving high precision through handcrafted methods and domain-specific knowledge, rule-based approaches have limitations. They are domain-dependent, unable to autonomously adapt, and involve extensive manual effort in crafting rules. For projects aiming to incorporate knowledge from diverse sources, the rigidity of rule-based approaches may pose challenges, as they confine input to specific parameters. As such, they are not the ideal option for the current project.

The ML approach involves the application of classification algorithms to automatically identify and categorize entities within text. They learn patterns and relationships from annotated training data, enabling them to generalize to unseen text. Examples of ML algorithms commonly applied include Support Vector Machines (SVM), which finds hyperplanes to separate entities in high-dimensional spaces, HMM, which is adept at handling sequential data like text, CRF, a type of probabilistic graphical model, and decision trees, which recursively split data based on feature values, making them suitable for capturing hierarchical relationships. These models do not rely on manually crafted rules, providing adaptability to different domains and languages. Moreover, they exhibit a capacity to generalize patterns learned from training data to identify entities in new, previously unseen text. The accuracy, however, may be constrained by the choice of classifier. Certain classifiers, such as HMM and SVM, may not adequately consider dependencies among words, limiting their effectiveness in capturing complex relationships within textual data.

The DL approach involves the application of neural networks, particularly deep neural architectures, to automatically learn hierarchical representations of features from raw textual data. Examples of DL architectures include RNNs, LSTMs, and transformer models like BERT. Advantages of the DL approach include its domain independence, allowing models to be less domain-dependant. The precision of inferred features is another notable advantage, as DL models can learn complex representations that enhance the accuracy of entity recognition. However, a key challenge associated with the DL approach is the need for a substantial amount of labeled data for training robust models. Deep neural networks, particularly large-scale models like BERT, require extensive datasets to learn meaningful representations and generalize effectively to diverse text. The usability of DL methods is therefore contingent on the availability and quality of the dataset used in the project.

Models' comparison

The majority of examined NLP articles, both NER and RE related, favored DL methodologies. The efficacy of these approaches is highest when complemented by a curated dataset and enough computational resources.

An examination of the NER models featured in the articles is encapsulated in Table 9. It reveals a prevalent trend among authors leveraging BERT or BERT-based models for the pre-training of data. Additionally, BiLSTM and CNN are commonly employed models for feature extraction in these contexts.

Table 9 - NER models' comparison

Reference	NER	RE	Model	F1-score (%)
[31]	X	X	BERT-PTHT	91,20
[28]	X		BERT_Sc	88,55
[26]	X		CNN	87,73
[30]	X	X	CNN-BERT	86,09
[49]	X	X	MEP	84,59
[13]	X		BERT-BiLSTM-IDCNN-CRF	81,18
[18]	X		BERT-BiLSTM-CRF	74,57
[19]	X		HMM-Rule	71,59
[17]	X		GRU	69,42

Similarly, a comparative analysis focusing on articles delving into RE, as is made in Table 10, underscores the frequent utilization of BERT and its variants. Moreover, a noteworthy inclusion in many instances is some form of the attention mechanism, indicative of its recurrent application in RE.

Table 10 - RE models' comparison

Reference	NER	RE	Model	F1-score (%)
[31]	X	X	BERT-PTHT	91,20
[23]		X	AT-BLSTM	86,30
[30]	X	X	CNN-BERT	86,09
[20]		X	BERT-BGRU-2ATT-CRF	85,40
[49]	X	X	MEP	84,59
[21]		X	Att-BLSTM	84,00
[11]		X	CNN+ATT+CNN+HATT	76,70
[33]		X	BERT-CRE	73,35
[12]		X	BERT-DistMult	72,24

Given the prominence of BERT in pretraining, BiLSTM and CNN in feature extraction, and CRF for corrections among the prevalent models, the current project will strategically adopt these methodologies. A comprehensive series of experiments and tests will be conducted to deepen the understanding of each model's nuances and ascertain their optimal integration within the project framework. This iterative process aims to refine the application of these models, ensuring their effectiveness and efficiency in achieving the project's objectives.

2.3.2 Learning Styles

The research also delved into the diverse learning styles exhibited by students. Various articles distinguished students based on either their preferred learning style or cognitive style [4]. As previously discussed, cognitive style refers to an individual's thought processes, rationalization, and decision-making in daily activities, whereas learning style encompasses the active methods individuals employ to acquire or absorb new information, whether in a classroom setting or when learning a new skill or language. It's worth noting that while these concepts are distinct, they are often interconnected [6].

As such, and to simplify the planning strategy, we'll treat them as interchangeable and consider different individual types equally valid in this study. As observed in numerous studies, a common distinction in people's learning styles is between analytical and intuitive learners. Analytical learners favor a systematic and logical approach to learning, thriving in structured environments with clear objectives, step-by-step instructions, and organized information. They excel in tasks that require attention to detail, precision in following instructions, and a methodical problem-solving approach. Conversely, intuitive learners rely on instinct, patterns, and a holistic understanding of information, thriving in creative, open-ended, and exploratory learning environments. They excel at recognizing patterns, making connections between concepts, and grasping overarching themes.

Keeping these learner differences in mind and aligning with the project's main objective, we've adopted an approach to display the output that accommodates both learner types, aiming to achieve the primary goal. A visual representation of the KG facilitates user understanding, with potential enhancements such as color coding to aid intuitive learners. To avoid overwhelming complexity, only the strongest and most relevant connections will be displayed. For analytical learners, detailed descriptions of each node will be generated, providing a clear explanation of the entity and its relationships.

Furthermore, as highlighted in [1], the incorporation of a chatbot emerges as a valuable addition. This feature would be instrumental in elucidating and addressing queries related to the generated KG and the underlying information sources. Consequently, it has been included as an optional objective for this project.

2.3.3 Similar Apps

In addition to the examined articles, a comprehensive analysis of various applications was conducted to survey the current state of the art. These applications are fully developed systems, which means that analyzing only the NER component would be insufficient. Instead, their exploration not only provided insights into existing practical applications but also served as a source of inspiration for defining the features to be incorporated into our own application. This section presents detailed analyses of these applications and culminates in a summary outlining the intended features drawn from these applications for implementation in our project.

Taskade[1]

Taskade functions as an AI-powered NLP tool designed to analyze raw text input, offering users a range of features, including summarization and mind mapping capabilities.

To assess its performance, a test was conducted using Taskade on the book "Harry Potter and the Philosopher's Stone" [50], where the tool was tasked with summarizing the story. The generated summary, presented in a chapter-wise bullet point format, exhibited accuracy issues, notably evident in consistency errors, as illustrated in Figure 2. For instance, the summary portrayed Harry discovering he is a wizard in chapter 2, contradicting the actual revelation by Hagrid in chapter 4.

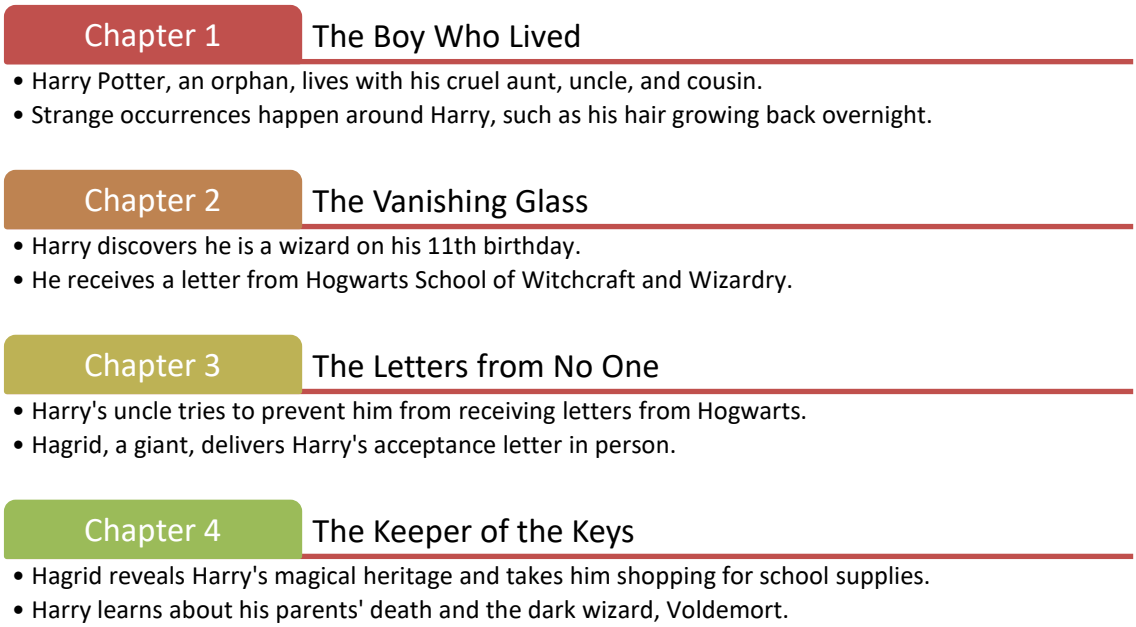

Figure 2 - Excerpt from the summarization of the first 4 chapters made in Taskade

Exploration of Taskade's mind map functionality revealed a chart featuring key entities categorized by the tool, such as “Main Characters”, “Plot Points” and “Key Locations”, which are to be expected, but also "Magical creatures," "Spells and Potions" which not typical in NLP analysis. Figure 3 displays a segment of the complete chart, exposing certain inaccuracies, such as inclusion of a quote from a different book ("After all this time? - Always") and references to concepts introduced in later books, like "Polyjuice Potion." This suggests that Taskade's mind map generation may involve information beyond the specific book provided as input, as the Harry Potter series is popularly used in the training of many NLP models.

[1] https://www.taskade.com/

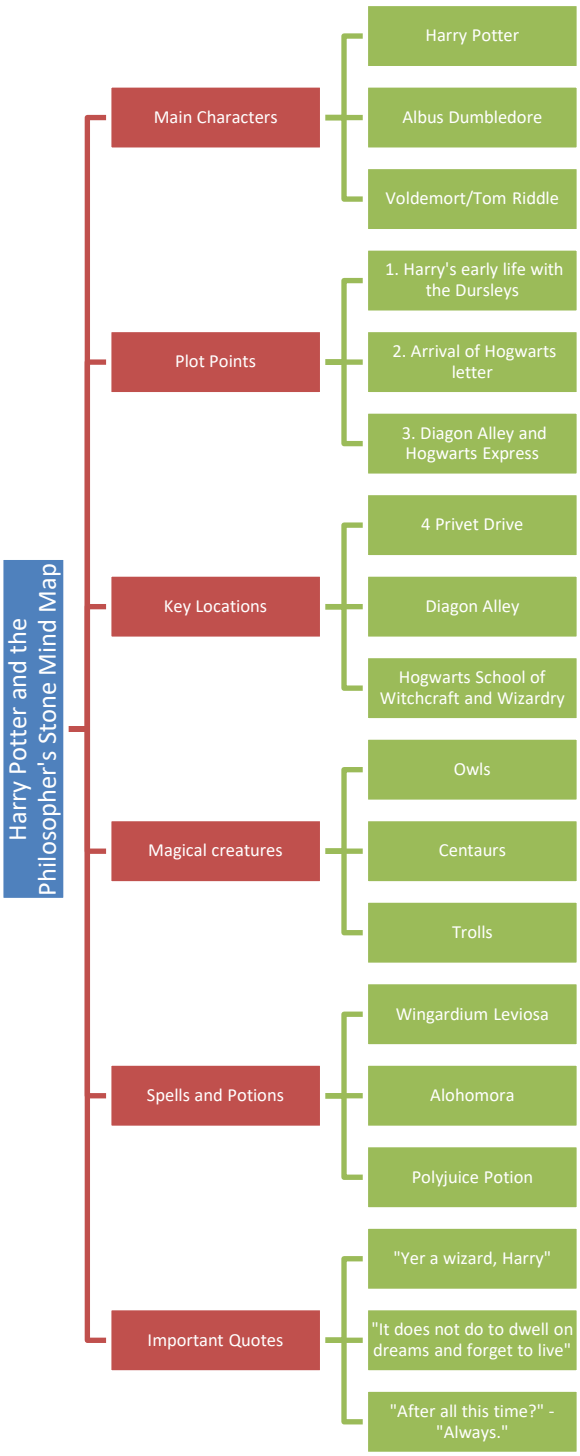

Figure 3 - Excerpt from the mind map made in Taskade

It is crucial to emphasize that the entities in the mind map were presented as named entries lacking content or relational connections, except for the previously mentioned categorization.

Furthermore, Taskade incorporated a chatbot feature, although its knowledge base seemed unrelated to the input provided, indicating an independent source of information.

MyReader[2]

MyReader is a chatbot designed to facilitate user interaction by allowing the upload and analysis of PDF files. It adopts a methodology that references the source when discussing specific sections, ensuring relevance to the user's questions. The chatbot limits its scope, abstaining from divulging external information unless expressly requested.

In a test scenario employing the tool with "Harry Potter and the Philosopher's Stone" [50] as input, the chatbot demonstrated a commitment to the provided source. When queried about events occurring in later books, the tool displayed a lack of knowledge, adhering strictly to the information within the given source. However, when insisted upon, the chatbot showcased its capability to draw from an external knowledge base, divulging additional information beyond the confines of the uploaded document, as depicted in Figure 4.

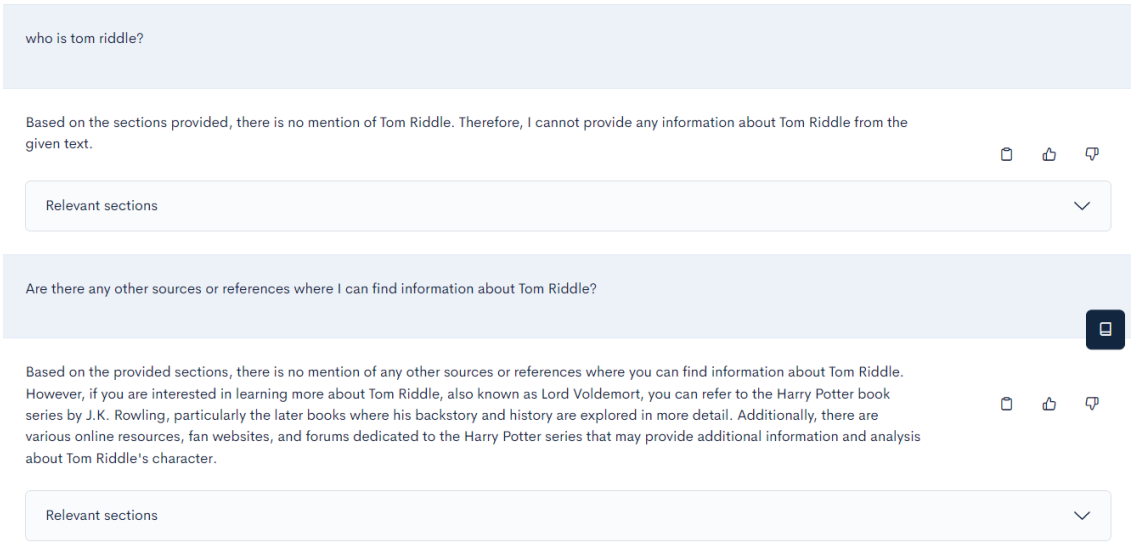

Figure 4 - Excerpt of outside source being used when insisted in MyReader

WordLift[3]

WordLift functions as a tool dedicated to website analysis, generating a KG encompassing its entities. Primarily employed for enhancing visibility in Google searches, this tool serves the purpose of optimizing content to better reach and engage with a targeted audience. The most prominent feature relevant to the current project is its named entity linking, establishing connections between entities on the website and known entries in databases. This linking imparts semantic meaning to each entity within the KG.

While the tool's potential significance aligns with the objectives of the current project, it's crucial to note that no testing was conducted due to WordLift being a paid tool.

[2] https://www.myreader.ai/
[3] https://wordlift.io

PopAI[4]

PopAI, akin to MyReader, functions as a chatbot specializing in processing PDF files as input. Its knowledge base is derived exclusively from the provided source, and it provides references to the source page that pertains to the queried topic. Notably, PopAI exhibits a limitation by confining its responses strictly to the content of the given text. Unlike its counterpart, it refrains from assuming or referencing information not explicitly mentioned in the provided text. However, PopAI is subject to a limitation in terms of document size, allowing only PDFs with a maximum of 50 pages. A test was conducted using the first 50 pages of "Harry Potter and the Philosopher's Stone" [50], encompassing approximately the initial 3 chapters. As depicted in Figure 5, the tool displayed a lack of knowledge about a character not yet introduced in the provided text.

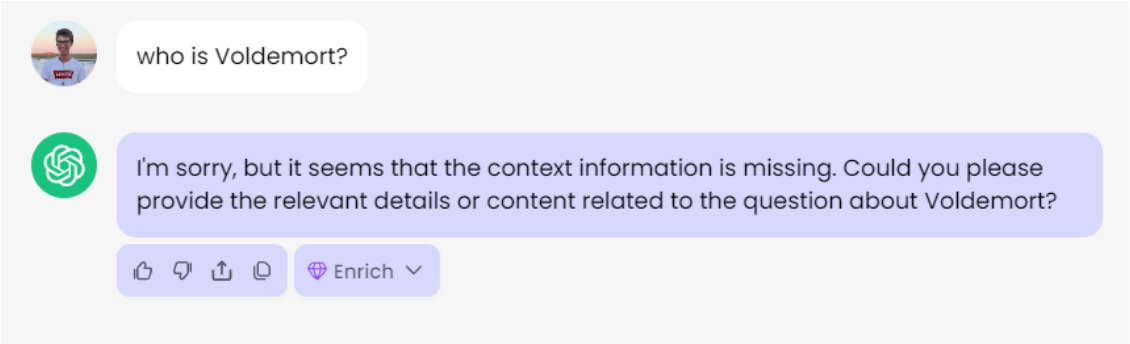

Figure 5 - Excerpt of conversation showcasing the lack of outside knowledge used in PopAI

Dandelion API[5]

Dandelion API serves as a versatile NLP Application Programming Interface (API) offering an array of features such as text similarity and sentiment analysis. The current focus is in its NER capabilities though. The API provides a demo where users can input text within a 700-character limit, and the API performs both NER and NEL. The API categorizes entities, although this categorization is based on metadata rather than linguistic analysis. It also provides a slider to adjust the extraction focus.

In comparing two tests conducted with text inputs related to the Mona Lisa and Harry Potter, as illustrated in Figure 6 and Figure 7 respectively, a notable distinction emerges. Leonardo da Vinci (a real entity) is categorized as a person, while Hermione Granger (a fictional entity) is categorized as a concept. This discrepancy underscores the API's ability to distinguish between real and fictional entities based on the provided text.

[4] https://www.popai.pro/chat/
[5] https://dandelion.eu/semantic-text/entity-extraction-demo/

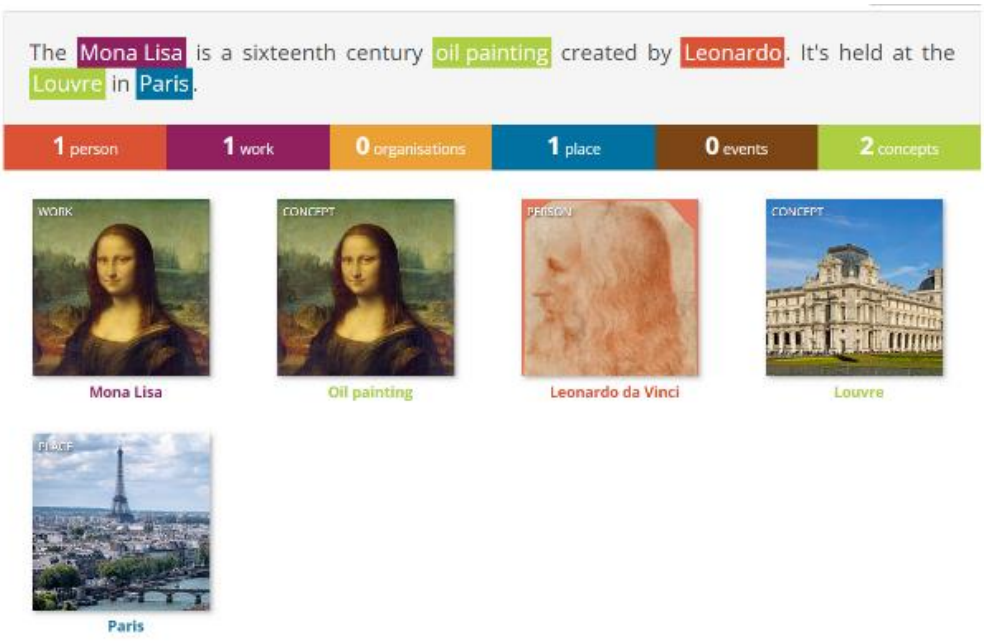

Figure 6 - Mona Lisa test in Dandelion API

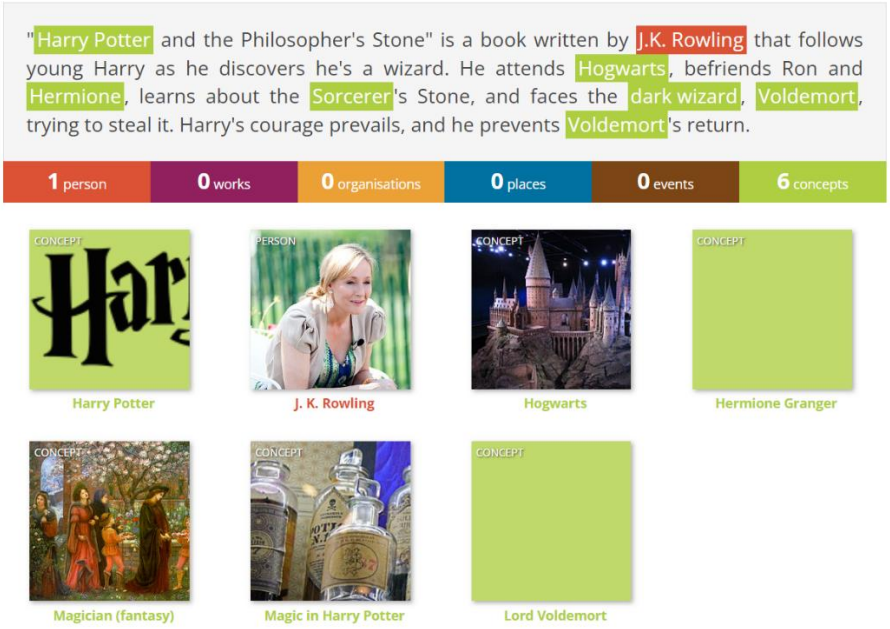

Figure 7 - Harry Potter test in Dandelion API

ChatPDF[6]

ChatPDF stands as another PDF-analysis chatbot within the realm of AI tools. In alignment with its counterparts, ChatPDF analyzes PDF files, extracting information and providing summaries. A distinctive characteristic, however, is its lack of references to the given text and lack of restrictions in referencing external knowledge. Unlike PopAI, which relies solely on source

[6] https://www.chatpdf.com/

information or MyReader which does the same unless explicitly prompted for external insights, ChatPDF freely draws upon external knowledge without constraints. Similarly to PopAI, ChatPDF imposes a page limit on the PDF files it processes. These negative aspects position it as the least robust among the three chatbot PDF tools.

During testing, the initial three chapters of "Harry Potter and the Philosopher's Stone" [50] were used as input, akin to PopAI. ChatPDF responds with knowledge pertaining to a character not introduced in the provided, which is evident in Figure 8.

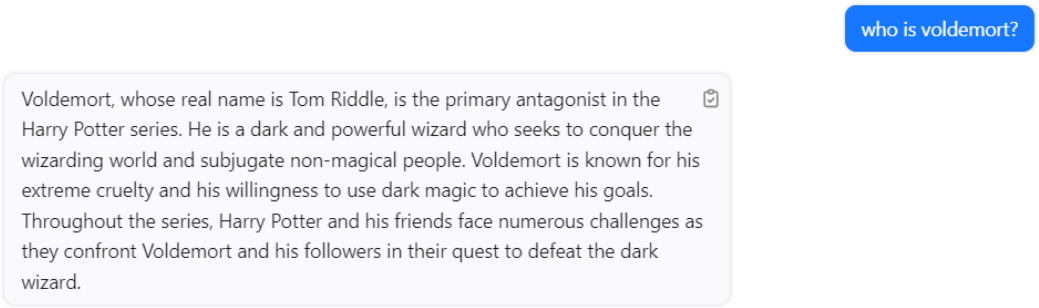

Figure 8 - Excerpt of conversation with ChatPDF

Conclusion

The assessed applications have showcased notable advancements in their respective domains. While each of them works very well in its designated functionalities, none precisely aligns with the specific objectives of this project. Nevertheless, valuable features can be distilled from each, as outlined in Table 11.

Table 11 - Similar Apps' features

	PDF input	Chatbot	NER	Bases KB in source only	References to position in source	Unlimited input count
Taskade	X	X	X			
MyReader	X	X		X	X	
WordLift			X			
PopAI	X	X		X	X	X
Dandelion API			X		X	X
ChatPDF	X	X				X

Taskade's mind map functionality is the closest visual representation to the envisioned future KG. However, the use of custom categories may cause confusion for users employing the

application with different sources. It is then decided to incorporate general common categories to enhance user clarity.

The NEL capability provided by Dandelion API proves to be a crucial feature for establishing relations between different entities within the KG. This form of RE represents a novel addition compared to the studied applications, setting it apart from them.

Among the three chatbots, PopAI stands out as the best due to its stringent adherence to the provided material. This alignment with the given source material is a crucial requirement for the project's success. However, a notable limitation is the imposed page limit, which limits the application’s usefulness.

Chapter 3

Methodology

This chapter outlines the systematic approach taken throughout the development and evaluation of the project. It begins by detailing the development process, showcasing the structured technique adopted to effectively tackle the challenges presented by the task. Next, the evaluation framework is discussed, focusing on the methods and metrics used to assess the performance of the model. Ethical considerations and data usage were also key aspects of this methodology, ensuring that the model was created with a strong focus on privacy, fairness, and transparency. Finally, the tools, applications, and frameworks employed during the project are highlighted, emphasizing their roles in supporting the successful development and deployment of the model.

3.1 Development

The Cross Industry Standard Process for Data Mining (CRISP-DM) has been adopted as the reference process model for the development and documentation of use cases in this dissertation. The CRISP-DM process comprises six distinct phases:

1. Business Understanding: Define project objectives and the business problem to be addressed.

2. Data Understanding: Gain insights into the dataset, seeking patterns, trends, and identifying any missing or irrelevant data.

3. Data Preparation: Prepare the data for modeling by cleaning, transforming, and selecting relevant data.

4. Modeling: Develop and test different models to find the most effective solution for the identified problem.

5. Evaluation: Evaluate the models to identify the most accurate and useful one against predefined criteria.

6. Deployment: Implement the chosen model for practical use.

By adhering to the CRISP-DM process, a structured and systematic approach is employed, ensuring comprehensive coverage of key aspects in the development and implementation of data mining solutions.

3.2 Evaluation

This section introduces the evaluation methods utilized in the project. In the project, the predictive models employ common evaluation techniques within the field of ML. The data utilized by these models is typically divided into training and testing sets. During the training phase, the model learns from the training data, and in the testing phase, it evaluates its predictive accuracy.

For instance, in the classification of entities within the input text into various categories, a confusion matrix is often employed to illustrate the performance of the model. The confusion matrix in Figure 9 outlines the following possibilities:

- True Positive (TP): Instances where the model correctly predicts a positive class (correctly classifies a node into a specific category).

- True Negative (TN): Instances where the model correctly predicts a negative class (correctly identifies a node as not belonging to a specific category).

- False Positive (FP): Instances where the model incorrectly predicts a positive class (erroneously classifies a node into a category).

- False Negative (FN): Instances where the model incorrectly predicts a negative class (fails to identify a node belonging to a specific category).

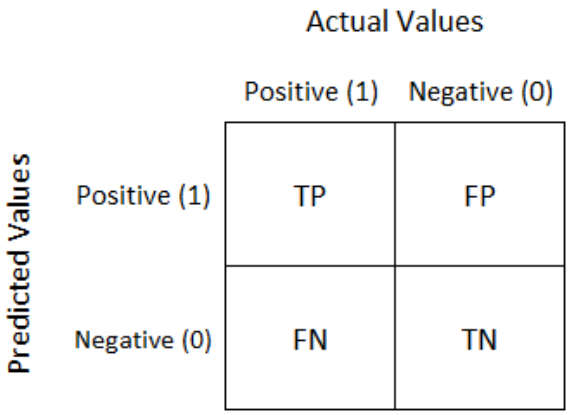

Figure 9 - Structure of Confusion Matrix

These values in the confusion matrix provide a comprehensive overview of the model's performance, allowing for a nuanced evaluation of its accuracy and effectiveness in classifying nodes within the KG. In the context of multiclass classification, where one class is considered positive and the rest are considered negative, various metrics can be derived from the confusion matrix to evaluate model performance.

Precision

Precision is the ratio of TP predictions to the total number of positive predictions. It measures the accuracy of the model when it predicts a positive class.

$$\boldsymbol{Precision = \frac{TP}{TP + FP}} \quad \textbf{(1)}$$

Recall (Sensitivity)

Recall, or sensitivity, is the ratio of TP predictions to the total number of actual positive instances. It assesses the model's ability to capture all instances of the positive class.

$$\boldsymbol{Recall = \frac{TP}{TP + FN}} \quad \textbf{(2)}$$

Specificity

Specificity is the ratio of TN predictions to the total number of actual negative instances. It measures the model's accuracy in predicting negative instances.

$$\boldsymbol{Specificity = \frac{TN}{TN + FP}} \quad \textbf{(3)}$$

Accuracy

Accuracy is the ratio of correct predictions (both TPs and TNs) to the total number of instances. It provides an overall measure of the model's correctness.

$$\boldsymbol{Accuracy = \frac{TP + TN}{TP + FN + TN + FP}} \quad \textbf{(4)}$$

F1-Score

The F1-Score is the harmonic mean of precision and recall. It provides a balanced measure that considers both FPs and FNs.

$$\boldsymbol{F_1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}} \quad \textbf{(5)}$$

3.3 Ethical Considerations and Data Usage

Throughout the development of this project, numerous ethical concerns were addressed, focusing particularly on privacy, fairness, transparency, and the responsible use of data. The goal was to ensure the highest ethical standards were maintained in both the design and deployment of the model.

Privacy and data protection were central considerations. The application was developed to process input data instantaneously and return results directly to the user without storing, sharing, or retaining any personal information. This approach was adopted to ensure that user privacy remains intact, mitigating potential data security risks. Despite not actively applying certain data protection measures, these aspects were heavily considered in the design to ensure they could be implemented if the system evolved for broader use or scaled applications.

In terms of fairness, particular care was taken in curating the dataset used for model training. Diverse data sources were chosen to ensure the model would operate effectively across various inputs and contexts, avoiding biases that could lead to unequal treatment of different user groups. While efforts were made to ensure fairness, it is acknowledged that ongoing monitoring and adjustments may be required as the tool encounters broader, more varied datasets.

Transparency was another significant ethical priority. Detailed documentation of the methodologies used, including model architecture and techniques, has been provided in this paper. This allows users to better understand how the model operates and how its predictions are generated. This level of openness helps to build trust in the system and provides a foundation for ethical scrutiny.

Regarding data usage, only labeled and analyzed datasets were employed for the training and fine-tuning of the BiLSTM-CRF model. Reputable platforms such as Kaggle, UCI Machine Learning Repository, Google Dataset Search, and AWS Public Datasets were utilized to source high-quality, annotated datasets focusing on key entity types like persons, organizations, and locations. These datasets were chosen not only for their reliability but also for their ethical handling and comprehensive coverage of diverse scenarios.

Beyond the current scope, the project considered broader ethical questions such as the potential misuse of NER technology and its implications for data privacy. While these concerns may not have been directly addressed in this implementation, they remain important for future applications of the model, particularly if it is expanded for more sensitive or large-scale deployments. The project strives to present an ethically acceptable tool with positive contributions and to minimize risks of harm in its output.

3.4 Tools and Frameworks

This section provides an overview of the technologies employed in developing this project. It commences with a discussion of external applications before delving into the programming language and libraries used throughout the development process.

3.4.1 External Applications

During this project, several external applications played pivotal roles, offering valuable assistance at various stages of the project development.

Rayyan was employed during the creation of the state-of-the-art section, specifically for the screening process of articles. Its functionality streamlined the selection of relevant articles for the study.

Mendeley Reference Manager proved instrumental in storing and organizing the selected articles used throughout the study. Its seamless integration with Microsoft Word facilitated the efficient insertion of references into the document created with Microsoft Word.

For the implementation of code, PyCharm served as the preferred integrated development environment (IDE). The institution provided a license for PyCharm to all its students, enabling a smooth and supported coding environment.

These applications collectively contributed to the efficiency and effectiveness of the project, offering specialized functionalities tailored to their respective roles in the research and development process.

3.4.2 Programming Languages and Libraries

The chosen programming language for this project is Python 3.10, a standard choice in the field. Despite the availability of Python 3.12 at the time of writing, we opted for a slightly older version to ensure compatibility with all frameworks and libraries. Several key libraries were utilized, including TensorFlow, PyTorch, and scikit-learn. These libraries offer a range of functions and pre-built models for tasks like data preprocessing, model training, and evaluation, enabling developers to focus on problem-solving rather than delving into low-level implementation details.

Python's extensive ecosystem includes a wealth of libraries for data visualization, manipulation, and analysis. Noteworthy among them are pandas, numpy, matplotlib, and seaborn, all of which play essential roles in ML tasks.

For the programming environment, Jupyter was selected. This open-source tool enables users to create and share documents featuring live code, data visualizations, and rich descriptions in Markdown. Widely embraced in the ML community, Jupyter facilitates data exploration, model prototyping, and the presentation of research findings.

Chapter 4

BERT-BiLSTM-CRF

This chapter reviews the design process of the NER model associated with the project. The main objective is to construct an entity extraction tool, and after comparing several models and approaches, it was determined that BERT in conjunction with a BiLSTM and a CRF correction layer would be the most suitable choice for entity recognition.

The chapter begins by introducing the selected dataset for the project and showcasing its details and specifications. It continues by exploring the project's model decision, detailing the advantages of using BERT, BiLSTM, and CRF. Next, the data preprocessing section describes the techniques used to prepare and curate the dataset. The encoding section discusses how BERT encodes the text data into a suitable format for the model. Following that, the training process outlines the setup and configuration for training the model. The evaluation metrics section discusses how the model's performance is measured. Finally, the design details provide a brief overview of the technical aspects involved in deploying the model.

4.1 Dataset

The dataset chosen for this NER task is the CoNLL-2003 dataset [51], a well-known labeled dataset specifically designed for NER. This dataset was selected after evaluating several alternatives, as it provides the necessary entity tags and is widely used in NER applications. The CoNLL-2003 dataset includes tags for organizations, people, and locations, which align with the objectives of this project. While it also contains POS tags, these were not relevant to the NER task and were only used early on but disregarded later. Additionally, the dataset includes files for German, but only the English data was used for the purposes of this project.

Table 12 presents key statistics of the CoNLL-2003 dataset, including the number of sentences, tokens, and each of the NER tags.

Table 12 - CoNLL-2003 dataset statistics

English data	Sentences	Tokens	LOC	MISC	ORG	PER
Training set	14,987	203,621	7140	3438	6321	6600
Validation set	3,466	51,362	1837	922	1341	1842
Test set	3,684	46,435	1668	702	1661	1617

One of the challenges encountered during the dataset preparation was forming complete sentences, as the dataset does not include explicit sentence separators. To address this issue, POS values were leveraged to identify end-of-sentence markers, such as '.', '!', or '?', ensuring that sentences were correctly delineated for further processing.

4.2 Model Architecture

As previously mentioned, the architecture chosen for the NER task consists of BERT, BiLSTM, and a CRF correction layer. This combination was selected due to its ability to effectively leverage contextual information and capture sequential dependencies in text. While some of the studied applications used CNNs alongside these models to improve feature extraction, this approach was unnecessary in the current context. The decision to use labeled data and to maintain a solid foundation and avoid unnecessary complexity was what guided this choice of streamlined architecture.

The BERT layer serves as the initial component of the architecture, processing raw text input into rich, contextualized word embeddings that consider the entire sentence's context for each word. This means that the same word with the same meaning can and will generate different embeddings in different sentences. This ability to capture semantic meaning is crucial for identifying entities in various contexts and avoid overfitting and it complements the bidirectional nature of the LSTM. The high-level functioning of the BERT embedding layer can be seen in Figure 10, which illustrates how BERT processes input text into contextualized embeddings.

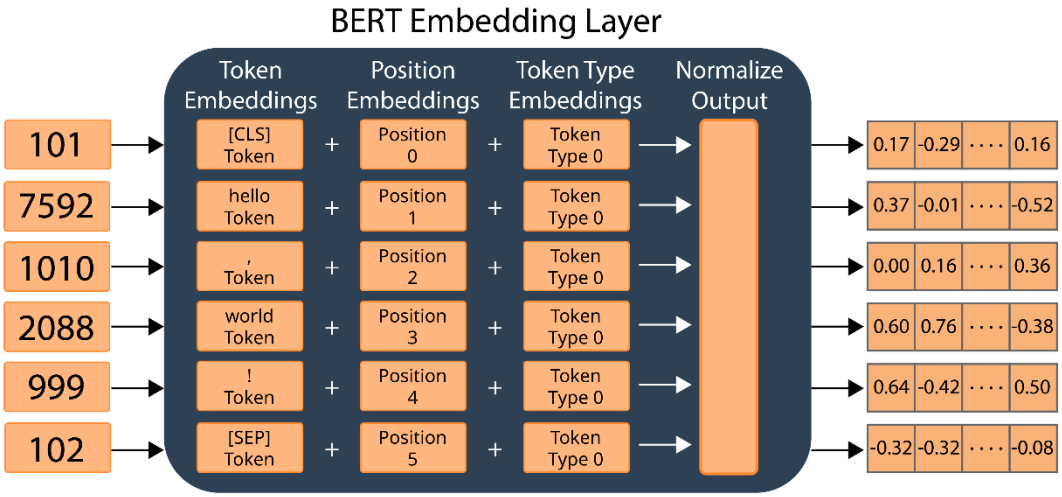

Figure 10 - BERT Embedding Layer

The BiLSTM layer follows, designed to capture the sequential nature of language by processing text in both directions—forward and backward. This bidirectional approach allows the model to consider both past and future words when predicting the current token's label, significantly enhancing its ability to understand entity boundaries and relationships in complex sentence structures.

Finally, the CRF layer is used to refine predictions by considering the relationships between consecutive labels, ensuring that the sequence of tags follows valid patterns. For example, an 'I-ORG' (inside organization) must always be preceded by a 'B-ORG' tag (beginning of organization). The CRF helps to enforce such constraints, improving the overall accuracy and consistency of the entity recognition process.

4.3 Data Preprocessing

The first step in data preprocessing involved a thorough analysis of the CoNLL-2003 dataset's composition. Each line in the dataset represents a token along with its corresponding tags. An initial examination revealed the presence of approximately 2,000 tokens labelled to be not a number (NaNs), which accounted for about 1% of the entire dataset; this was determined to be a non-issue, and these tokens were subsequently removed.

Following this, the label counts were assessed to identify any imbalances in the dataset, which would be relevant for the classification task within the BiLSTM model.

To facilitate model training, the labels were indexed, converting them into numerical values. An additional label was included to represent tokens that needed to be ignored, such as special tokens or padding.

To address the lack of explicit sentence boundaries in the dataset, a manual inspection was conducted to discern a pattern for identifying the ends of sentences. Utilizing the POS tags,

which consistently marked the end of a sentence, the dataset was effectively segmented into individual sentences.

It is important to note that the CoNLL-2003 dataset comes pre-split into training, validation, and test sets. Consequently, all preprocessing and other future operations were uniformly applied across these three subsets. While this section outlines the primary steps taken in preprocessing, further processing related to BERT will be discussed in the subsequent Encoding section.

4.4 Encoding

The encoding process for the dataset begins with an examination of the longest sentence to ensure it falls within the limits of BERT's token capacity, which is capped at 512 tokens. Following this, the BERT tokenizer is loaded, specifically using the bert-base-cased model, a widely adopted pretrained model from Hugging Face. This model was chosen due to its effective handling of case sensitivity, which is critical for NER tasks where capitalization can indicate important distinctions—such as identifying entities like people, locations, and organizations, which are typically capitalized. While both the cased and uncased versions were tested for tokenization, the cased model was ultimately selected as the most appropriate for this project.

During tokenization, special tokens—CLS and SEP—are added at the beginning and end of each sentence, respectively, to mark the boundaries of the input for BERT. In this process, any sentences exceeding 512 tokens are pruned, which was found to be less than 0.5% of the total sentences in the dataset, thus confirming that this was not a significant issue.

One complication encountered with BERT is its handling of unrecognized words, which are often split into smaller sub word units for tokenization. This splitting can misalign the labels, necessitating a manual check to ensure that the correct labels are duplicated for each sub word, maintaining the integrity of the entity tagging.

After tokenization, the datasets are split into batches of equal length, and segment IDs are generated. While these segment IDs are required by BERT for distinguishing between different sentences, in this case, they do not significantly impact the process. Attention masks are also created, and everything is padded to the maximum length of 512 tokens. For the data format, PyTorch tensors are utilized, as they are the standard input required by BERT and are supported by popular libraries.

Once the dataset is properly formatted, the three subsets (training, validation, and test) are fed through the pretrained BERT encoder, which transforms each token into 768-dimensional embeddings. Here, the attention masks play a crucial role in differentiating between actual tokens and padding tokens.

To highlight the contextual ability of BERT’s encoder an example was made where the same word with the same meaning was placed in two different sentences and fed through the encoder model. The produced embeddings rely on the entire sentence’s context, resulting in distinct embeddings for the same word in each sentence, as shown in Table 13.

Table 13 - Example of Contextual BERT Embedding

Sentence	Word	Embedding (First 3 Dimensions)
"**This** is a test sentence."	This	[0.024, -0.234, 1.262, ...]
"**This** is quite possibly the wrong room."	This	[0.461, -0.157, 0.866, ...]

A notable challenge during this phase is the lengthy runtime associated with generating the embeddings. To optimize this process, it was decided to store the embeddings in a file, allowing for faster loading instead of re-running the encoding process each time the environment is launched.

Upon generating or loading the embeddings, validation involves checking the shapes to ensure consistency with the original number of sentences and tokens, as well as assessing their mean values and standard deviations to confirm they fall within reasonable limits.

Following this encoding process, the next step involves applying the BiLSTM-CRF model to the encoded embeddings.

4.5 Training Process

After generating the BERT embeddings, the model was constructed, comprising a BiLSTM layer, a dropout layer, a linear layer, and a CRF correction layer. The BiLSTM layer processes the embeddings in both forward and backward directions, capturing sequential dependencies. A dropout layer is included to prevent overfitting by randomly dropping units during training. The linear layer maps the hidden states to tag space, followed by the CRF layer, which calculates the loss based on the predicted tags, improving sequence labelling by considering label dependencies.

Initially, the model was trained using baseline hyperparameters to establish a point of comparison. As seen in Table 14, these hyperparameters included a learning rate of 0.001, with the Adam optimizer used for training. The hidden dimension was set to 256, allowing for sufficient capacity to learn from the dataset. A dropout rate of 0.3 was applied to reduce overfitting, and no weight decay was added at this stage to keep the configuration simple. Class weights were also not applied during the initial runs to observe how the model performed without accounting for class imbalance.

Table 14 - Base Hyperparameters

Parameter	Value
Learning Rate	0.001
Hidden Dimension	256
Dropout Rate	0.3
Number of Epochs	5
Optimizer	Adam
Weight Decay	0
Training Batch Size	56
Validation Batch Size	156
Test Batch Size	60
Use Class Weight	False

The model was trained for 5 epochs to gather early feedback on its trends before diving into more extensive tuning. While Stochastic Gradient Descent (SGD) was briefly tested as an alternative optimizer, no significant improvements were observed, leading to its exclusion in favor of Adam for subsequent experiments.

Later, class weights were introduced to address the dataset's class imbalance. These weights were computed using Additive Smoothing with a constant of $\alpha = 10^4$, according to Formula 6:

$$\boldsymbol{w_i = \frac{Total\ Frequency + n \cdot \alpha}{n \cdot (Frequency_i + \alpha)}} \quad \textbf{(6)}$$

where w_i represents the weight for class i, n is the total number of classes, and $Frequency_i$ is the observed frequency of class i. The smoothing constant α ensures no class is weighted disproportionately. The “IGNORE” class (used for padding) was assigned a weight of zero to exclude it from the model’s learning process.

Switching the training to the GPU halved the training time compared to CPU-based runs, enabling faster exploration of hyperparameter configurations and more efficient iteration cycles.

Training and validation losses were tracked during each run to monitor model performance. As shown in Figure 11, both losses decreased throughout training with minimal divergence, indicating overfitting was well-controlled, likely due to the dropout layer.

Figure 11 - Training and Validation Loss in Base Hyperparameters

However, the training loss drops more sharply, from around 100 to near 0, while the validation loss decreases from 450 to around 300. This difference can be attributed to multiple factors: the sizes of the batches and the datasets themselves.

The training dataset was larger, providing the model with more examples to learn from, which generally improves generalization and results in a more effective decrease in training loss. Additionally, the smaller training batch size allows for more frequent updates to the model weights, contributing to a smoother drop in training loss.

In contrast, the validation dataset was smaller, which can lead to less reliable estimates of model performance, resulting in a steeper curve and a higher overall validation loss. The larger validation batch size averages out the loss over fewer examples, making the curve appear less responsive.

4.6 Evaluation Metrics

In a multiclass NER system, predicting TN does not provide meaningful insights, as the negative class encompasses all other labels in the classification. Consequently, this project focuses on metrics that measure TP, ensuring a clearer evaluation of the model's performance. The primary evaluation metrics used are precision, recall, and the F1 score, with assessments conducted on both the validation set and a test dataset employing the most prominent parameters.

Precision measures the accuracy of a model's positive predictions, calculated as the ratio of true positives to the sum of true positives and false positives. In a multiclass NER system, high precision means that most of the predicted named entities are correct, minimizing the number of incorrect labels. For example, if a model predicts five entities—three correctly labelled as "Location" (true positives) and two incorrectly labelled (false positives)—the precision would be $Precision = \frac{3}{3+2} = 0.6$. This indicates that while the model identified several entities, not

all were accurate, highlighting the importance of high precision to ensure that predictions are reliable.

Recall evaluates the model's ability to identify all relevant instances within the dataset, calculated as the ratio of true positive predictions to the sum of true positives and false negatives. In a multiclass NER system, high recall indicates that the model has successfully identified most of the actual named entities present in the text. For example, if the phrase "New York" appears in the input but the model fails to recognize it, the recall score will decrease, reflecting a missed opportunity to capture relevant information.

The F1 Score combines precision and recall into a single metric that provides a balance between the two, offering a more comprehensive measure of the model's performance. A higher F1 score indicates a better balance between precision and recall, making it particularly useful for evaluating NER systems where both false positives and false negatives can be detrimental. In this project, the F1 score is vital in ensuring that the model not only identifies entities accurately but also captures as many relevant entities as possible.

Figure 12 presents an example of the performance metrics, illustrating the trends in precision, recall, and F1 score throughout the training process, using the base hyperparameters.

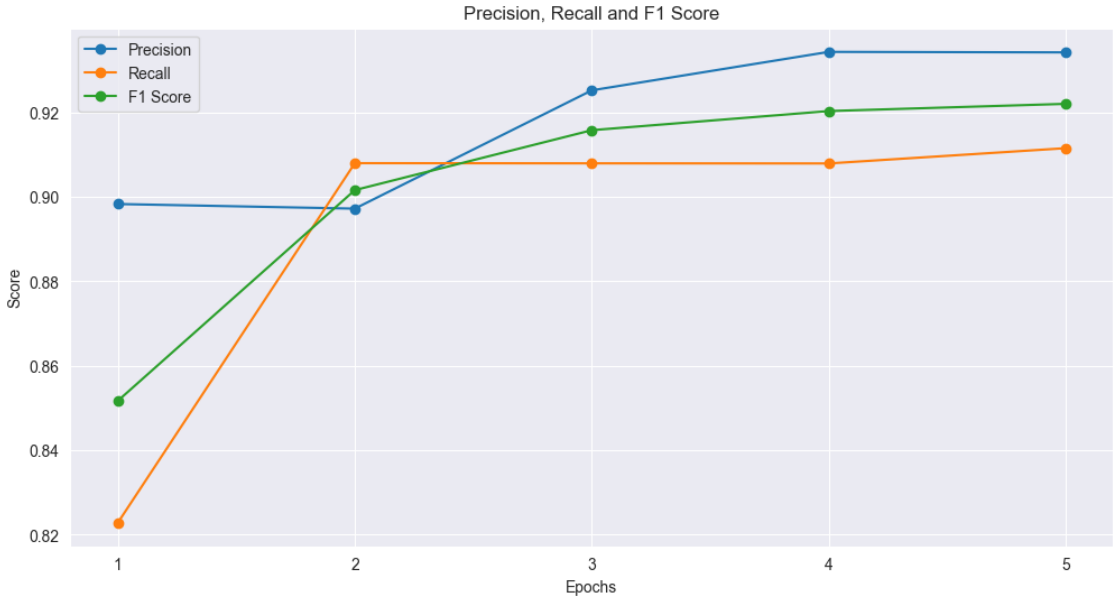

Figure 12 - Metrics for Base Hyperparameters

Initially, precision is high while recall is low, indicating that the model is conservative in its predictions, only attempting to identify a few entities but doing so accurately. As training progresses, recall rises while precision slightly decreases, suggesting that the model is now identifying more entities, albeit with some mistakes. Following this, recall stabilizes, meaning the model maintains a consistent effort in predicting entities, while its precision improves, leading to a notable increase in correct predictions. This dynamic is reflected in the gradual rise of the F1 score, which effectively balances precision and recall throughout the entire training process.

As observed, relying on only one of these metrics would not provide a complete picture of the model's performance. Precision alone overlooks missed entities, while recall fails to account for the accuracy of predictions. Thus, it is essential to evaluate these metrics collectively and complement them with qualitative assessments for a more comprehensive understanding of the model's effectiveness.

4.7 Design Details

This project utilizes several key tools and frameworks to facilitate the development and deployment of the NER model. The libraries employed include JSON for data handling, Matplotlib for visualization, and Pandas for data manipulation. The BERT model is accessed through the Hugging Face Transformers library, using the bert-base-cased variant. For data preprocessing and sequence padding, TensorFlow's Keras API is utilized. PyTorch serves as the primary deep learning framework, with additional support from the TorchCRF library for the CRF layer and various optimization and evaluation metrics, including precision, recall, and F1 score.

Large files for storing embeddings were created using HDF5 to minimize the need for rerunning the BERT model for every execution. The project was developed in PyCharm using Python 3.10, with Anaconda to manage the environment. Version control was maintained using Git, allowing for a streamlined development process. The hardware setup for this project is shown in Table 15.

Table 15 - Project Hardware Specifications

Component	Specification
Processor	AMD Ryzen 7 5700X 8-Core
Video Card	NVIDIA GeForce RTX 3070 Ti
Operating System	Windows 11
RAM	32 GB

Chapter 5

Results

This chapter outlines the findings of the model's performance in NER tasks, focusing on the influence of hyperparameter tuning on key metrics like precision, recall, and F1 scores. The evaluation combines quantitative analyses and qualitative assessments from manual testing to gauge the model's effectiveness in identifying entities.

It begins with a discussion of the comparison strategy used to assess model performance across various configurations, followed by an overview of the hyperparameter settings explored. The chapter concludes with an error analysis that highlights specific misclassifications, shedding light on the model's strengths and limitations. Together, these insights provide a comprehensive view of the model's performance, and the efficacy of the tuning strategies applied.

5.1 Comparison Strategy

The model's performance was evaluated using both training and validation loss as general metrics, primarily to gauge overall performance and detect overfitting. However, precision, recall, and F1 scores were prioritized due to their relevance in handling the positive class. In NER, where there are multiple classes and class imbalance, these metrics give a more insightful measure of the model's effectiveness, particularly in recognizing named entities accurately.

Several hyperparameters were defined, as mentioned previously, with their initial values either based on typical averages observed in studied articles [13], [20], [30], and relevant research found online, or chosen for simplicity, where simplicity is defined as reducing the number of model steps during training. For instance, class weights were not initially used to avoid unnecessary complexity. After defining the base hyperparameters, more options were explored

for each parameter. Typically, one value was chosen above and one below the baseline, though in certain cases, two values were selected toward one end of the spectrum.

Each parameter was then evaluated by altering only one value at a time, while keeping the rest of the parameters constant. This resulted in a total of 20 parameter options, meaning the model was trained 20 times with different configurations.

This strategy allowed for a clear comparison between the baseline model and models where only one parameter differed. While this approach provided valuable insights into how each parameter influenced the model, ideally, this process would be repeated multiple times to minimize the impact of random outliers. However, due to time and resource constraints, such repetitions were not performed, as the potential benefit did not outweigh the effort required.

For each metric (training loss, validation loss, precision, recall, F1), values were plotted in a graph where the current model's performance was shown alongside the base model. For precision, recall, and F1 score, additional global-scale graphs were created to visualize the results across all configurations, avoiding overcrowding individual graphs while providing a broader view of how each metric compared across different setups. This allowed an initial comparison of the base model with each configuration and gave a clearer understanding of how the metrics evolved with each parameter change. An example of a comparison screen can be seen in Figure 13:

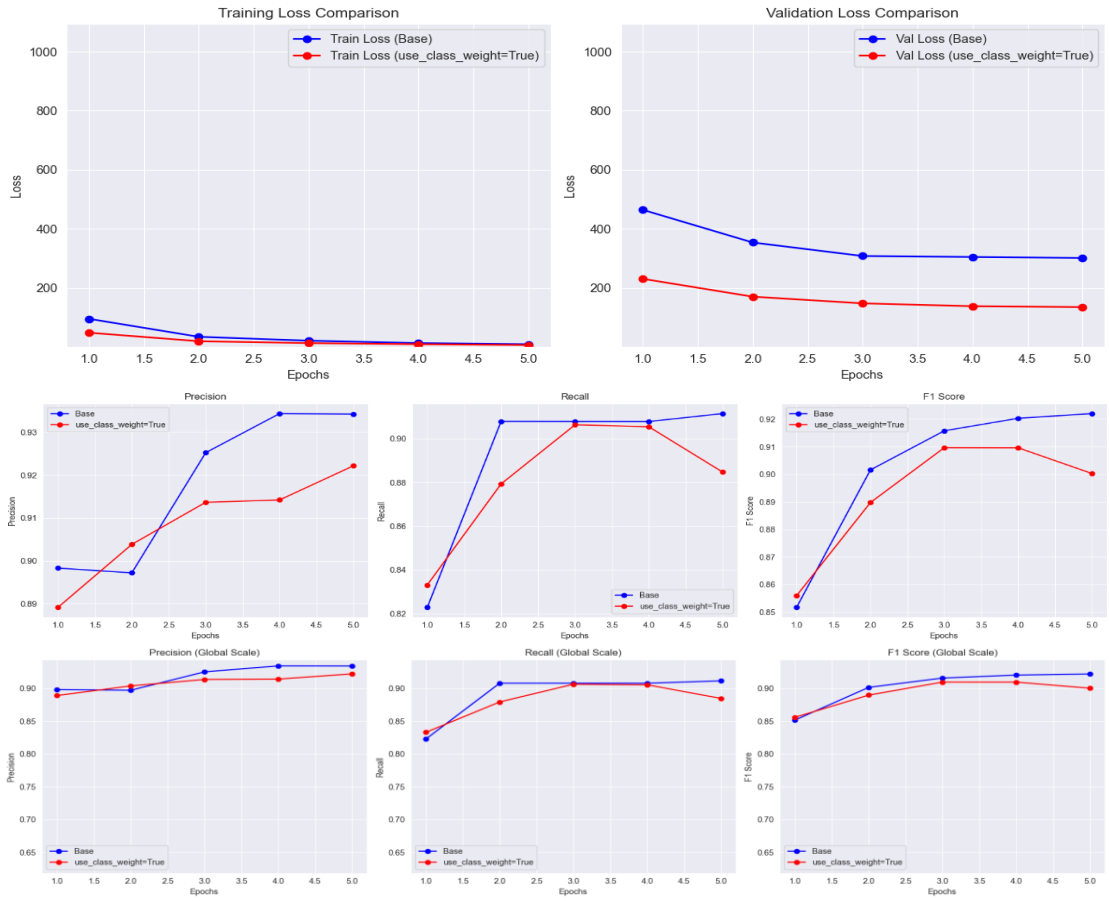

Figure 13 - Comparison screen of model metrics

5.2 Hyperparameter Configurations

To assess the impact of various hyperparameters on the model's performance, a set of base parameters was established alongside their potential alternatives. Table 16 presents the base hyperparameters along with the explored configurations.

Table 16 - Hyperparameters' possible values

Parameter	Base Value	Possible Value 1	Possible Value 2
Learning Rate	0.001	0.0005	0.0001
Hidden Dimension	256	128	512
Dropout Rate	0.3	0.1	0.5
Number of Epochs	5	10	20
Optimizer	Adam	N/A	
Weight Decay	0	1×10^{-5}	1×10^{-4}
Training Batch Size	56	14	28
Validation Batch Size	156	N/A	
Use Class Weights	False	True	

Only the Adam optimizer was used, as it is a widely recognized baseline known for its efficiency in training deep learning models; this choice helped avoid additional complexity that might obscure the assessment of other hyperparameter impacts. Additionally, the validation batch size remained constant, as altering it was expected to have minimal influence on the model's learning dynamics. A consistent validation batch size facilitates stable evaluation metrics, ensuring reliability in validation performance throughout the training process.

5.2.1 Learning Rate

The learning rate started at 0.001 and was reduced to 0.0005 and 0.0001. While the lower values resulted in a decrease in training loss, this reduction occurred at a slower rate than with

the base parameters. Consequently, the validation loss was generally higher compared to the base model, indicating that a lower learning rate may hinder convergence speed. Additionally, precision was lower with the reduced rates, while recall started lower but improved more significantly, leading to an overall lower F1 score. In conclusion, the learning rate of 0.001 (base) allows for faster convergence and superior performance compared to lower rates. Although lower learning rates result in steadier recall improvement, they slow down precision gains, negatively impacting overall effectiveness.

5.2.2 Hidden Dimension

The hidden dimension was first tested at 128 compared to the base of 256. The training loss was very similar to the base model, while the validation loss also mirrored this trend, albeit with a slight increase in the 5th epoch. With a hidden dimension of 128, there were noticeable fluctuations in precision. Both precision and recall, along with F1 score, peaked at epoch 4, surpassing the base model, but then dipped in the 5th epoch, leading to lower values overall. This suggests that a hidden dimension of 128 may lack the expressive capacity necessary for stable performance, indicating that a larger hidden dimension can better handle the complexity of the task.

When the hidden dimension was increased to 512, the model demonstrated faster convergence and achieved a slightly lower final training loss (9.04 compared to 7.68), with both models starting around 90. Although the validation loss began better for the model with a hidden dimension of 512, the fluctuations suggest instability associated with the larger size. Ultimately, the final validation loss for the 512-dimension model was slightly higher than that of the 256-dimensional model. Despite the initial advantages in precision, recall, and F1 score, the final results for the 512-dimension model were slightly lower than those for the 256-dimensional model. This indicates that while a higher hidden dimension can promote quicker convergence, it may introduce greater variability in performance metrics and validation loss.

Increasing the hidden dimension enhances the model's capacity to capture complex patterns, as more neurons can improve expressive power and stability. A smaller hidden dimension, like 128, can lead to fluctuations and reduced performance. However, excessively increasing the hidden dimension can cause overfitting or instability, as seen with the 512-dimensional model, which, despite a strong start, ultimately yielded less consistent results. Therefore, choosing the appropriate hidden dimension is essential for balancing model complexity and stability.

5.2.3 Dropout rate

The model with a dropout rate of 0.1 achieves a lower final validation loss and better recall but experiences more fluctuations and lower precision and F1 scores compared to a dropout rate of 0.3. The latter provides greater stability and a better balance among precision, recall, and F1 metrics.

The model with a dropout rate of 0.5 shows fluctuations in precision and recall throughout the training process, ultimately achieving better precision and a lower validation loss than the 0.3 rate. However, it exhibits slightly lower recall and F1 scores, indicating instability, as results could vary with additional epochs.

A dropout rate of 0.3 appears to offer the best balance between training and validation losses, providing consistent performance across all metrics. Lower dropout rates may improve recall and prevent overfitting but sacrifice precision and stability, while higher rates lead to improved final validation loss but introduce fluctuations, highlighting the need to optimize dropout for effective regularization.

5.2.4 Number of Epochs

Increasing the training duration to 10 epochs results in a lower training loss, reflecting better optimization; however, it also raises the risk of overfitting. The final validation loss for the 10-epoch model is higher, indicating potential overfitting after the 5th epoch. Although recall remains largely unchanged, with a slight increase for the 10-epoch model, both precision and F1 scores decrease, suggesting a decline in overall performance. Unless some extra strategies are implemented, stopping at 5 epochs may be more effective for this model to maintain balance and prevent overfitting.

5.2.5 Weight Decay

Introducing weight decay initially raises the training loss, though it converges similarly to the base model. The validation loss remains slightly higher with weight decay, indicating it might not be aiding generalization. Precision tends to be lower when weight decay is applied, while recall shows a marginal improvement toward the end but remains close to the base model. F1 scores are also slightly reduced when using weight decay. The base model without weight decay performs better, particularly in precision and validation loss, suggesting better generalization without weight decay.

5.2.6 Training Batch Size

Reducing the batch size leads to higher train and validation losses, indicating slower convergence and poorer generalization. Precision is lower with the smaller batch size, while recall fluctuates more but remains similar. The F1 score is slightly reduced, showing that overall performance is not as good as the base model. Overall, a base model with a batch size of 56 offers better results.

5.2.7 Use Class Weights

Applying class weights significantly reduces both training and validation losses, indicating improved handling of class imbalances and better generalization. Precision remains high but is slightly lower than the base model, while recall improves early on but doesn't sustain the advantage over time. The F1 score stays strong but ends slightly lower than the base model, suggesting a trade-off between precision and recall. Overall, class weights help reduce losses and improve generalization but come with a minor compromise in the balance between precision and recall.

5.3 Tuning the Base Model

After the study of the different parameters and their effects on the model's performance, it was time to select specific parameters to modify the base model for fine-tuning. Table 17 summarizes the final hyperparameter values compared to the original settings:

Table 17 – Comparison between final and original hyperparameters

Parameter	Base Value	Possible Value 1
Learning Rate	0.001	0.001
Hidden Dimension	256	256
Dropout Rate	0.3	0.2
Number of Epochs	5	15
Optimizer	Adam	Adam
Weight Decay	0	1×10^{-5}
Training Batch Size	56	56
Validation Batch Size	156	156
Use Class Weights	False	True

The number of epochs was increased from 5 to 15 to provide additional training time for the model, which was expected to enhance learning, although it required careful monitoring to avoid overfitting. To combat overfitting, class weights were enabled to address class imbalance, which was critical for improving recall and F1 scores, especially in a multi-class NER context. Weight decay was introduced at a value of 1e-5 to regularize the model, improving precision and F1 scores while slightly reducing recall. Finally, the dropout rate was reduced from 0.3 to 0.2 to help stabilize validation loss and further enhance model performance.

These adjustments collectively contributed to fine-tuning the model, balancing key performance metrics while addressing potential overfitting issues. Following these modifications, the final configuration yielded a satisfactory performance, aligning with the study's objectives. The metrics for the final model were evaluated using the test dataset, providing an objective assessment of its performance. The results are summarized in Table 18.

Table 18 - Final Model's Metrics

Metric	Value
Precision	0.8825
Recall	0.8646
F1 Score	0.8722

These metrics indicate a balanced effectiveness in recognizing the positive class while maintaining a solid overall performance, thereby reflecting the successful adjustments made during the fine-tuning process.

5.4 Error Analysis

The performance of each class was measured using the test dataset, with precision, recall, and F1 score calculated for each class. These values are displayed in Figure 14. Additionally, the relative frequency of each class within the entities—excluding the 'O' class—was calculated, and this frequency is represented in the chart within the figure.

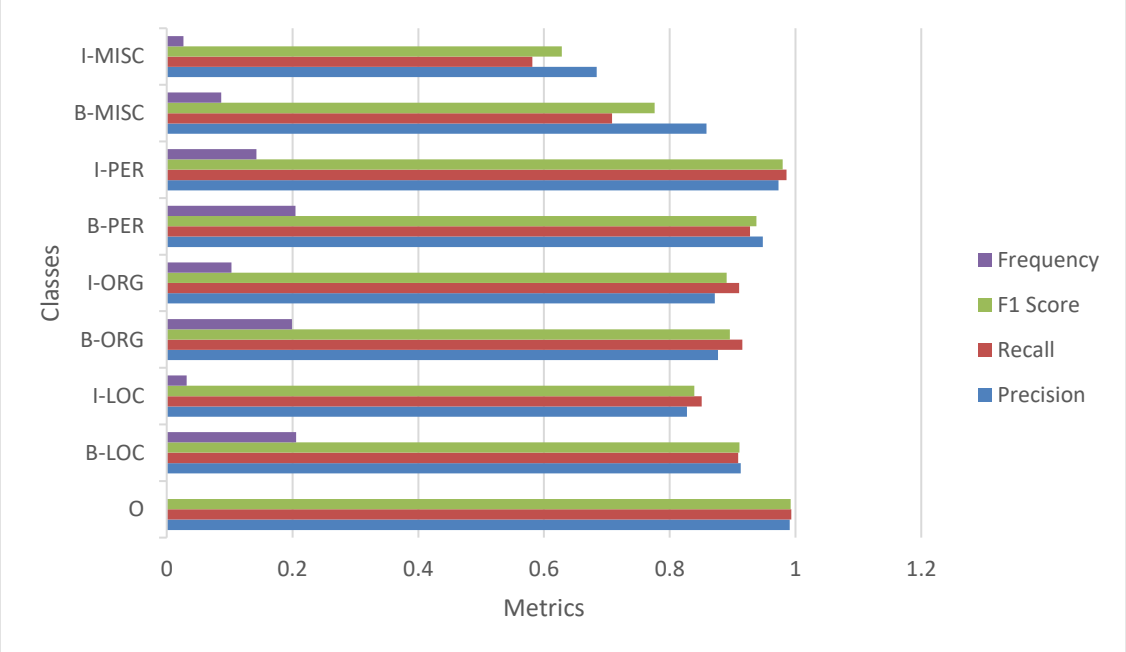

Figure 14 - Test Dataset Class Metrics

A connection can be observed between the lower frequency of MISC entities and their corresponding lower values of precision, recall, and F1 score. The lower score for Organization entities, compared to Location entities, is likely due to a small portion of tokens being common between the two labels. It is also noted that, within the test dataset, entities comprise about 18% of the tokens. Despite this, there is not a significant discrepancy between the metrics for

entity classes and those for the non-entity class ('O'), which demonstrates both the quality of the model, and the effectiveness of the techniques applied.

In the manual testing process, only a few errors were identified, with one notable example illustrated in the sentence "Taylor Swift buys vegetables in the USA," the model successfully predicted the correct labels. "Taylor Swift" was identified as a person and "USA" as a location, which are the expected predictions:

Taylor	Swift	buys	vegetables	in	the	USA.
B-PER	**I-PER**	**O**	**O**	**O**	**O**	**B-LOC**

The introduction of an organization, such as "Walmart," caused the model to incorrectly interpret "Taylor" as the start of an organization, followed by "Swift" being labelled as a person. This prediction is logically impossible, as an entity cannot simultaneously be part of both categories.

Taylor	Swift	buys	vegetables	at	Walmart	in	the	USA.
B-ORG	**I-PER**	**O**	**O**	**O**	**B-ORG**	**O**	**O**	**B-LOC**

The final change presents a curious case. When using "Tailor Swift" instead of "Taylor Swift," the model correctly predicted the labels once again. This is particularly interesting given that "Tailor" is predominantly recognized as a profession rather than a common first name, yet the model correctly identifies it in this context.

Tailor	Swift	buys	vegetables	at	Walmart	in	the	USA.
B-PER	**I-PER**	**O**	**O**	**O**	**B-ORG**	**O**	**O**	**B-LOC**

This illustrates both the strengths of the model in certain contexts and its limitations in handling specific cases involving entity overlap or ambiguity.

Chapter 6

Conclusion

This chapter summarizes the key outcomes of the project, reflecting on the performance of the implemented NER model and comparing it to state-of-the-art alternatives. It revisits the project's objectives, discussing both the accomplishments and limitations encountered. Suggestions for future work are presented, outlining potential areas for improvement and expansion of the model. Finally, a reflection on the personal academic growth achieved during the course of this project is provided.

6.1 Model Comparison

The NER model implemented in this project consists of a pretrained BERT encoder integrated with a BiLSTM model and a CRF correctional layer. The final results achieved by this model can be compared with other NER models discussed in the state-of-the-art chapter, as summarized in Table 19.

Table 19 - Comparison of results with other models

Reference	NER	RE	Model	F1-score (%)
[31]	X	X	BERT-PTHT	91,20
[28]	X		BERT_Sc	88,55
[26]	X		CNN	87,73
[30]	X	X	CNN-BERT	86,09
[49]	X	X	MEP	84,59
[13]	X		BERT-BiLSTM-IDCNN-CRF	81,18
[18]	X		BERT-BiLSTM-CRF	74,57
[19]	X		HMM-Rule	71,59
[17]	X		GRU	69,42
-	X		BERT-BiLSTM-CRF (this project)	87.22

Analysing all models, it can be concluded that the current model performs favourably alongside other successful models. This demonstrates the chosen method's efficacy and validates its application in the context of named entity recognition.

6.2 Project Accomplishments

The objectives set for this project are worth revisiting. The primary goal of constructing an entity extraction tool was successfully achieved, along with the secondary objectives tied to it. It is important to note that the initial aim was broader, targeting the construction of a tool capable of generating knowledge graphs by using both Named Entity Recognition (NER) and Relation Extraction (RE). The research conducted aligns well with these original objectives. However, due to time constraints, the scope of the project was restructured to focus solely on NER, a more manageable target within the given timeframe.

Despite this shift, the current project remains highly relevant in the fields of AI, NER, and NLP. A functional entity extraction tool with strong performance has been developed, providing a solid foundation for future work, including the potential development of a knowledge graph tool to enhance learning.

6.3 Future Work

For future work, the model's errors identified in the Results chapter should be addressed to enhance overall performance. Training the model with additional or more diverse datasets could improve its capabilities and increase its range of entity recognition. Further hyperparameter tuning may also result in performance gains. Implementing automatic early stopping could streamline experimentation by allowing the effects of different configurations to be studied more efficiently. Additionally, experimenting with different optimizers or class weight functions may yield beneficial changes. A new model could also be created and tested, or novel techniques such as IDCNNs could be applied to the current model. This technique can help capture long-range dependencies more effectively, making them useful in enhancing sequential tasks like NER.

Beyond enhancing the current model, future work should focus on achieving the project's original objectives. This includes applying RE techniques to the NER output to identify relationships between recognized entities [24]. Exploring methods such as active learning for schema expansion [24], along with integrating BERT and CNNs for capturing relations [15], could be beneficial. Additionally, employing entity linking and clustering techniques may help refine connections between entities [9]. The ultimate goal is to develop a knowledge graph tool, using graph visualizers [31], that effectively combines NER and RE, facilitating the construction of comprehensive KGs from raw input. Implementing this tool would significantly advance AI applications in education.

6.4 Final Reflections

Overall, this project can be considered a success, even if it did not fully reach the original scope. The developed model is capable of identifying most entities effectively, and the project has contributed valuable research by comparing different NER systems and demonstrating the effectiveness of a specific approach.

On a personal level, the project fostered both academic growth and technical development. It also led to personal growth, particularly in terms of increased self-awareness, as well as improved resource and time management skills.

References

[1] I. Ozturk, “The Role of Education in Economic Development: A Theoretical Perspective,” *SSRN Electronic Journal*, 2008, doi: 10.2139/ssrn.1137541.

[2] Global Education Monitoring Report Team, *Global Education Monitoring Report 2020: Inclusion and education: All means all. Paris*. UNESCO, 2020. doi: 10.54676/JJNK6989.

[3] K. Gyimah-Brempong, O. Paddison, and W. Mitiku, “Higher education and economic growth in Africa,” *J Dev Stud*, vol. 42, no. 3, pp. 509–529, Apr. 2006, doi: 10.1080/00220380600576490.

[4] A. Iku-Silan, G.-J. Hwang, and C.-H. Chen, “Decision-guided chatbots and cognitive styles in interdisciplinary learning,” *Comput Educ*, vol. 201, p. 104812, Dec. 2023, doi: https://doi.org/10.1016/j.compedu.2023.104812.

[5] K. Dobashi, C. P. Ho, C. P. Fulford, M.-F. Grace Lin, and C. Higa, “Learning pattern classification using moodle logs and the visualization of browsing processes by time-series cross-section,” *Computers and Education: Artificial Intelligence*, vol. 3, p. 100105, Dec. 2022, doi: https://doi.org/10.1016/j.caeai.2022.100105.

[6] X. Gu *et al.*, “Active versus Passive Strategy in Online Creativity Training: How to Best Promote Creativity of Students with Different Cognitive Styles?,” *Think Skills Creat*, vol. 44, p. 101021, Dec. 2022, doi: https://doi.org/10.1016/j.tsc.2022.101021.

[7] E. H. Sara, Z. Rajae, and R. J. Idrissi, “Pedagogical innovation on interactive graphic animations: Case study of synaptic transmission - 1st year baccalaureate degree, life and earth sciences, Morocco,” *Social Sciences & Humanities Open*, vol. 3, no. 1, p. 100103, Dec. 2021, doi: https://doi.org/10.1016/j.ssaho.2020.100103.

[8] C. Thomas, K. A. V Puneeth Sarma, S. Swaroop Gajula, and D. B. Jayagopi, “Automatic prediction of presentation style and student engagement from videos,” *Computers and Education: Artificial Intelligence*, vol. 3, p. 100079, Dec. 2022, doi: https://doi.org/10.1016/j.caeai.2022.100079.

[9] Q. Huang, Z. Yuan, Z. Xing, Z. Zuo, C. Wang, and X. Xia, “1+1$>$2: Programming Know-What and Know-How Knowledge Fusion, Semantic Enrichment and Coherent Application,” *IEEE Trans Serv Comput*, vol. 16, no. 3, pp. 1–14, 2022, doi: 10.1109/TSC.2022.3207273.

[10] Y. Hu, H. Shen, W. Liu, F. Min, X. Qiao, and K. Jin, “A Graph Convolutional Network With Multiple Dependency Representations for Relation Extraction,” *IEEE Access*, vol. 9, pp. 81575–81587, 2021, doi: 10.1109/ACCESS.2021.3086480.

[11] Z. Zhu, J. Su, and Y. Zhou, “Improving Distantly Supervised Relation Classification With Attention and Semantic Weight,” *IEEE Access*, vol. 7, pp. 91160–91168, 2019, doi: 10.1109/ACCESS.2019.2925502.

[12] W. Hong, S. Li, Z. Hu, A. Rasool, Q. Jiang, and Y. Weng, “Improving Relation Extraction by Knowledge Representation Learning,” in *2021 IEEE 33rd International Conference on Tools with Artificial Intelligence (ICTAI)*, IEEE, Nov. 2021, pp. 1211–1215. doi: 10.1109/ICTAI52525.2021.00191.

[13] Y. Chang, L. Kong, K. Jia, and Q. Meng, “Chinese named entity recognition method based on BERT,” in *2021 IEEE International Conference on Data Science and Computer Application (ICDSCA)*, IEEE, Oct. 2021, pp. 294–299. doi: 10.1109/ICDSCA53499.2021.9650256.

[14] K. Bhattacharjee *et al.*, “Named Entity Recognition: A Survey for Indian Languages,” in *2019 2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT)*, IEEE, Jul. 2019, pp. 217–220. doi: 10.1109/ICICICT46008.2019.8993236.

[15] T. Al-Moslmi, M. Gallofre Ocana, A. L. Opdahl, and C. Veres, “Named Entity Extraction for Knowledge Graphs: A Literature Overview,” *IEEE Access*, vol. 8, pp. 32862–32881, 2020, doi: 10.1109/ACCESS.2020.2973928.

[16] A. Anandika and S. P. Mishra, “A Study on Machine Learning Approaches for Named Entity Recognition,” in *2019 International Conference on Applied Machine Learning (ICAML)*, IEEE, May 2019, pp. 153–159. doi: 10.1109/ICAML48257.2019.00037.

[17] N. Banik and Md. H. H. Rahman, “GRU based Named Entity Recognition System for Bangla Online Newspapers,” in *2018 International Conference on Innovation in Engineering and Technology (ICIET)*, IEEE, Dec. 2018, pp. 1–6. doi: 10.1109/CIET.2018.8660795.

[18] Z. Dai, X. Wang, P. Ni, Y. Li, G. Li, and X. Bai, “Named Entity Recognition Using BERT BiLSTM CRF for Chinese Electronic Health Records,” in *2019 12th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*, IEEE, Oct. 2019, pp. 1–5. doi: 10.1109/CISP-BMEI48845.2019.8965823.

[19] M. D. Drovo, M. Chowdhury, S. I. Uday, and A. K. Das, “Named Entity Recognition in Bengali Text Using Merged Hidden Markov Model and Rule Base Approach,” in *2019 7th International Conference on Smart Computing & Communications (ICSCC)*, IEEE, Jun. 2019, pp. 1–5. doi: 10.1109/ICSCC.2019.8843661.

[20] J. Lv, J. Du, N. Zhou, and Z. Xue, “BERT-BIGRU-CRF: A Novel Entity Relationship Extraction Model,” in *2020 IEEE International Conference on Knowledge Graph (ICKG)*, IEEE, Aug. 2020, pp. 157–164. doi: 10.1109/ICBK50248.2020.00032.

[21] M. Shi, J. Huang, and C. Li, “Entity Relationship Extraction Based on BLSTM Model,” in *2019 IEEE/ACIS 18th International Conference on Computer and Information Science (ICIS)*, IEEE, Jun. 2019, pp. 266–269. doi: 10.1109/ICIS46139.2019.8940185.

[22] T. Chen, N. Wang, M. He, and L. Sun, “Reducing Wrong Labels for Distantly Supervised Relation Extraction With Reinforcement Learning,” *IEEE Access*, vol. 8, pp. 81320–81330, 2020, doi: 10.1109/ACCESS.2020.2990680.

[23] R. Zhang, F. Meng, Y. Zhou, and B. Liu, “Relation classification via recurrent neural network with attention and tensor layers,” *Big Data Mining and Analytics*, vol. 1, no. 3, pp. 234–244, Sep. 2018, doi: 10.26599/BDMA.2018.9020022.

[24] S. Seo *et al.*, “Active Learning for Knowledge Graph Schema Expansion,” *IEEE Trans Knowl Data Eng*, vol. 34, no. 12, pp. 5610–5620, Dec. 2022, doi: 10.1109/TKDE.2021.3070317.

[25] A. J. Bingham, J. F. Pane, E. D. Steiner, and L. S. Hamilton, “Ahead of the Curve: Implementation Challenges in Personalized Learning School Models,” *Educational Policy*, vol. 32, no. 3, pp. 454–489, May 2018, doi: 10.1177/0895904816637688.

[26] J. Wang, F. Song, K. Walia, J. Farber, and R. Dara, “Using Convolutional Neural Networks to Extract Keywords and Keyphrases: A Case Study for Foodborne Illnesses,” in *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*, IEEE, Dec. 2019, pp. 1398–1403. doi: 10.1109/ICMLA.2019.00228.

[27] E.-O. Bosse, J. Falardeau, I. Prevost, E. Shahbazian, and O. Labonte, “Domain Specific Fusion of Unstructured Text for Situation Understanding (Poster),” in *2019 22th International Conference on Information Fusion (FUSION)*, IEEE, Jul. 2019, pp. 1–6. doi: 10.23919/FUSION43075.2019.9011243.

[28] L. Akhtyamova, P. Martinez, K. Verspoor, and J. Cardiff, “Testing Contextualized Word Embeddings to Improve NER in Spanish Clinical Case Narratives,” *IEEE Access*, vol. 8, pp. 164717–164726, 2020, doi: 10.1109/ACCESS.2020.3018688.

[29] A. Madan and U. Ghose, “Sentiment Analysis for Twitter Data in the Hindi Language,” in *2021 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, IEEE, Jan. 2021, pp. 784–789. doi: 10.1109/Confluence51648.2021.9377142.

[30] W. Deng and Y. Liu, “Chinese Triple Extraction Based on BERT Model,” in *2021 15th International Conference on Ubiquitous Information Management and Communication (IMCOM)*, IEEE, Jan. 2021, pp. 1–5. doi: 10.1109/IMCOM51814.2021.9377404.

[31] F. B. Rodrigues, W. F. Giozza, R. de Oliveira Albuquerque, and L. J. Garcia Villalba, “Natural Language Processing Applied to Forensics Information Extraction With

Transformers and Graph Visualization," *IEEE Trans Comput Soc Syst*, pp. 1–17, 2022, doi: 10.1109/TCSS.2022.3159677.

[32] Y. Wang, Y. Sun, Z. Ma, L. Gao, Y. Xu, and Y. Wu, "A Method of Relation Extraction Using Pre-training Models," in *2020 13th International Symposium on Computational Intelligence and Design (ISCID)*, IEEE, Dec. 2020, pp. 176–179. doi: 10.1109/ISCID51228.2020.00046.

[33] J. Hou, X. Li, H. Yao, H. Sun, T. Mai, and R. Zhu, "BERT-Based Chinese Relation Extraction for Public Security," *IEEE Access*, vol. 8, pp. 132367–132375, 2020, doi: 10.1109/ACCESS.2020.3002863.

[34] T. Sun, C. Zhang, Y. Ji, and Z. Hu, "MSnet: Multi-Head Self-Attention Network for Distantly Supervised Relation Extraction," *IEEE Access*, vol. 7, pp. 54472–54482, 2019, doi: 10.1109/ACCESS.2019.2913316.

[35] M. L. Bernacki, M. J. Greene, and N. G. Lobczowski, "A Systematic Review of Research on Personalized Learning: Personalized by Whom, to What, How, and for What Purpose(s)?," *Educ Psychol Rev*, vol. 33, no. 4, pp. 1675–1715, Dec. 2021, doi: 10.1007/s10648-021-09615-8.

[36] Z. Pajalic, "Reflections on acquired university teaching skills gathered over 20 years at Swedish and Norwegian universities," *Social Sciences & Humanities Open*, vol. 8, no. 1, p. 100650, Dec. 2023, doi: https://doi.org/10.1016/j.ssaho.2023.100650.

[37] M. Durnali, "'Destroying barriers to critical thinking' to surge the effect of self-leadership skills on electronic learning styles," *Think Skills Creat*, vol. 46, p. 101130, Dec. 2022, doi: https://doi.org/10.1016/j.tsc.2022.101130.

[38] R. H. C. Machado, S. V. Conceição, R. Pelissari, S. Ben Amor, and T. L. Resende, "A multiple criteria framework to assess learning methodologies," *Think Skills Creat*, vol. 48, p. 101290, Dec. 2023, doi: https://doi.org/10.1016/j.tsc.2023.101290.

[39] I. Ramis-Conde and A. Hope, "Training teachers in maintaining equity in the micro-moments of a mathematical dialogue," *Teach Teach Educ*, vol. 87, p. 102924, Dec. 2020, doi: https://doi.org/10.1016/j.tate.2019.102924.

[40] L. Lin, P. Ginns, T. Wang, and P. Zhang, "Using a pedagogical agent to deliver conversational style instruction: What benefits can you obtain?," *Comput Educ*, vol. 143, p. 103658, Dec. 2020, doi: https://doi.org/10.1016/j.compedu.2019.103658.

[41] G. Zhu and C. A. Iglesias, "Exploiting semantic similarity for named entity disambiguation in knowledge graphs," *Expert Syst Appl*, vol. 101, pp. 8–24, Jul. 2018, doi: 10.1016/j.eswa.2018.02.011.

[42] J. L. Martinez-Rodriguez, I. Lopez-Arevalo, and A. B. Rios-Alvarado, “OpenIE-based approach for Knowledge Graph construction from text,” *Expert Syst Appl*, vol. 113, pp. 339–355, Dec. 2018, doi: 10.1016/j.eswa.2018.07.017.

[43] P. Fafalios, M. Baritakis, and Y. Tzitzikas, “Exploiting Linked Data for Open and Configurable Named Entity Extraction,” *International Journal on Artificial Intelligence Tools*, vol. 24, no. 02, p. 1540012, Apr. 2015, doi: 10.1142/S0218213015400126.

[44] S. Zenasni, E. Kergosien, M. Roche, and M. Teisseire, “Spatial Information Extraction from Short Messages,” *Expert Syst Appl*, vol. 95, pp. 351–367, Apr. 2018, doi: 10.1016/j.eswa.2017.11.025.

[45] X. Ren, A. El-Kishky, C. Wang, F. Tao, C. R. Voss, and J. Han, “ClusType,” in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, NY, USA: ACM, Aug. 2015, pp. 995–1004. doi: 10.1145/2783258.2783362.

[46] M. B. HABIB and M. VAN KEULEN, “TwitterNEED: A hybrid approach for named entity extraction and disambiguation for tweet,” *Nat Lang Eng*, vol. 22, no. 3, pp. 423–456, May 2016, doi: 10.1017/S1351324915000194.

[47] M. Fossati, E. Dorigatti, and C. Giuliano, “N-ary relation extraction for simultaneous T-Box and A-Box knowledge base augmentation,” *Semant Web*, vol. 9, no. 4, pp. 413–439, Jun. 2018, doi: 10.3233/SW-170269.

[48] Z. Xu, X. Luo, S. Zhang, X. Wei, L. Mei, and C. Hu, “Mining temporal explicit and implicit semantic relations between entities using web search engines,” *Future Generation Computer Systems*, vol. 37, pp. 468–477, Jul. 2014, doi: 10.1016/j.future.2013.09.027.

[49] J. Chen and J. Gu, “Jointly Extract Entities and Their Relations From Biomedical Text,” *IEEE Access*, vol. 7, pp. 162818–162827, 2019, doi: 10.1109/ACCESS.2019.2952154.

[50] J. K. Rowling, *Harry potter and the philosopher’s stone*. Bloomsbury Childrens Books, 2014.

[51] E. F. Tjong, K. Sang, and F. De Meulder, “Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition,” 2003, Accessed: Sep. 28, 2024. [Online]. Available: http://lcg-www.uia.ac.be/conll2003/ner/