NOVA SCHOOL OF
SCIENCE & TECHNOLOGY

DEPARTMENT OF
COMPUTER SCIENCE

RODRIGO MIGUEL GAMEIRO VILHENA GONÇALVES

BSc in Computer Science And Technology

ENTITY RECOGNITION AND LINKING FOR BIOMEDICAL DOCUMENTS

APPLYING RECENT TRANSFORMER-BASED ENTITY RECOGNITION AND LINKING ALGORITHMS FOR THE BIOMEDICAL DOMAIN, TO A MULTI-LINGUAL SCENARIO

MASTER IN COMPUTER SCIENCE AND ENGINEERING

NOVA University Lisbon
September, 2024

DEPARTMENT OF
COMPUTER SCIENCE

ENTITY RECOGNITION AND LINKING FOR BIOMEDICAL DOCUMENTS

APPLYING RECENT TRANSFORMER-BASED ENTITY RECOGNITION AND LINKING ALGORITHMS FOR THE BIOMEDICAL DOMAIN, TO A MULTI-LINGUAL SCENARIO

RODRIGO MIGUEL GAMEIRO VILHENA GONÇALVES

BSc in Computer Science And Technology

Adviser: André Lamúrias
Assistant Professor, NOVA School of Science and Technology

MASTER IN COMPUTER SCIENCE AND ENGINEERING

NOVA University Lisbon
September, 2024

Entity Recognition and Linking for Biomedical Documents

Applying recent Transformer-based Entity Recognition and Linking Algorithms for the Biomedical Domain, to a Multi-Lingual Scenario

This document was created with the (pdf/Xe/Lua)LaTeX processor and the NOVAthesis template (v7.1.27) (Lourenço, 2021).

To my family.

Acknowledgements

First of all, I would like to thank my advisor Prof. André Lamúrias for his guidance and availability. His insightful feedback and patience have shaped the direction and quality of this dissertation.

To my friends, both from home and from university, for welcoming me into their lives and being by my side for years. Their friendship, companionship, and laughter provided me with some much needed moments of relief from any challenges I faced, no matter how daunting.

To my girlfriend, who I would like to thank for her patience, kindness and understanding, for never letting me fall, and for always being there for me when I needed it the most. Her love and support was immeasurable and indispensable.

I would also like to thank my brother for his encouragement and belief in me, that have always been one of my primary sources of motivation. His constant vigilance has also been a reassuring presence in my life.

Lastly, to my parents, I would like to express my deepest gratitude for their unconditional love and unwavering support, not only throughout my academic journey, but through all aspects of my life. I am forever grateful for the sacrifices they've made to provide me with the opportunity to grow into the person I am today.

"You just can't beat the person who never gives up."

— Babe Ruth

Abstract

In the ever expanding domain of Biomedicine, a wealth of crucial information is embedded within an extensive number of free-text documents. However, the unstructured nature of this textual reservoir, coupled with the intricacy of biomedical terminology, poses a significant challenge for automated systems to extract valuable insights in an efficient way. Furthermore, the majority of resources are allocated to English, but other lower-resource languages also contain valuable information. The current state-of-the-art models in multilingual biomedical Natural Language Processing lag behind their general domain and English-specific counterparts. This disparity emphasizes the need for approaches that can tackle the complexity of biomedical literature across languages.

This dissertation focuses on two Natural Language Processing tasks within the information extraction realm: Named Entity Recognition (NER) and Entity Linking (EL). To tackle these tasks, we used the following methodology: first, we investigated the efficacy of transfer learning approaches, by adapting pre-trained Transformer-based models to the complexities of multilingual biomedical texts. Second, we employed a data augmentation technique to enrich the training data and try to enhance the performance of those models. We evaluated our approaches on the SympTEMIST, CANTEMIST and MultiCardioNER shared tasks - competitions that provide a benchmark for evaluating NER and EL techniques within the biomedical domain for various languages. We obtained competitive results for both NER and EL, consistently surpassing the mean and median results for those shared tasks, and even establishing a new state-of-the-art score for DrugTEMIST English in NER. Our methodology can be easily extended to other languages and datasets.

Keywords: Natural Language Processing, Biomedicine, Multilingual, Transformers, Transfer Learning, Data Augmentation

Resumo

Num domínio em constante expansão como a Biomedicina, grande parte da informação encontra-se em documentos de texto livre. No entanto, a natureza não estruturada deste reservatório textual, juntamente com a complexidade da terminologia biomédica, representa um desafio significativo na tentativa de extrair informação valiosa de forma eficiente por parte de sistemas automatizados. Para além disso, a maioria dos recursos estão disponíveis em inglês, mas outras línguas com menos recursos também possuem informações que podem ser bastante valiosas. O actual estado de arte dos modelos de Processamento de Linguagem Natural (em inglês *Natural Language Processing*) no âmbito biomédico e num cenário multilingue fica aquém dos modelos sem domínio específico e focados na língua inglesa. Esta disparidade realça a necessidade de desenvolver abordagens que possam lidar com a complexidade da literatura biomédica em várias línguas.

Esta dissertação centra-se em duas tarefas de Processamento de Linguagem Natural, no âmbito da extração de informação: Reconhecimento de Entidades Nomeadas (*Named Entity Recognition*, ou NER) e Mapeamento de Entidades (*Entity Linking*, EL). Para abordar estas tarefas, utilizámos a seguinte metodologia: primeiro, investigámos a eficácia de abordagens de aprendizagem por transferência (*Transfer Learning*), adaptando modelos pré-treinados baseados na arquitetura *Transformer* às complexidades dos textos biomédicos na vertente multilingue. Segundo, desenvolvemos uma técnica de aumento de dados (*Data Augmentation*) para enriquecer os dados de treino e tentar melhorar o desempenho destes modelos. Avaliámos as nossas abordagens nas tarefas partilhadas (*shared tasks*) SympTEMIST, CANTEMIST e MultiCardioNER - competições que se dedicam à avaliação de técnicas de NER e EL, no domínio biomédico em várias línguas. Obtivemos resultados competitivos tanto para NER como para EL, ultrapassando consistentemente os resultados médios e medianos das tarefas partilhadas referidas, e estabelecendo um novo estado de arte, nomeadamente para DrugTEMIST *English* em NER. A nossa metodologia pode ser facilmente estendida a outras línguas e conjuntos de dados.

Palavras-chave: Processamento de Linguagem Natural, Biomedicina, Multilingue, *Transformers*, Aprendizagem por Transferência, Aumento de Dados

Contents

List of Figures x

List of Tables xi

Glossary xiv

Acronyms xvi

1 Introduction 1

1.1 Motivation 1

1.2 Objectives 2

1.3 Contributions 3

1.4 Document Structure 4

2 Related Work 6

2.1 Natural Language Processing 6

2.1.1 Definition 6

2.1.2 Brief History 6

2.2 Attention Mechanism and Transformers 8

2.2.1 Input Preprocessing Steps 8

2.2.2 Attention 9

2.2.3 The Transformer Architecture 13

2.3 Large Language Models 16

2.3.1 Pre-Training and Fine-Tuning 16

2.3.2 Types of Large Language Models 17

2.4 Named Entity Recognition 21

2.4.1 Definition and Motivation 22

2.4.2 Evaluation 22

2.4.3 Techniques 24

2.4.4 Challenges 24

2.5 Entity Linking 25
2.5.1 Definition and Motivation 25
2.5.2 Entity Linking In Combination With Named Entity Recognition . 25
2.5.3 Knowledge bases 26
2.5.4 Evaluation 27
2.5.5 Techniques 27
2.5.6 Challenges 28
2.6 Transfer Learning and Data Augmentation 30
2.6.1 Transfer Learning 30
2.6.2 Data Augmentation 31
2.7 Shared Tasks 31
2.7.1 SympTEMIST 32
2.7.2 CANTEMIST 34
2.7.3 MultiCardioNER 34

3 Methodology and Experiments **37**
3.1 Third-Party Tools 37
3.2 Data 38
3.2.1 SympTEMIST 38
3.2.2 CANTEMIST 40
3.2.3 MultiCardioNER 40
3.3 Named Entity Recognition Experiments 41
3.3.1 Transfer Learning 42
3.3.2 Data Augmentation 51
3.4 Entity Linking Experiments 57
3.4.1 Transfer Learning 58
3.4.2 Data Augmentation 66

4 Results and Discussion **71**
4.1 Named Entity Recognition Results 71
4.1.1 Transfer Learning Results 71
4.1.2 Data Augmentation Results 75
4.1.3 Best Named Entity Recognition Results 81
4.2 Entity Linking Results 82
4.2.1 Transfer Learning Results 82
4.2.2 Data Augmentation Results 85
4.2.3 Best Entity Linking Results 90

5 Conclusion **92**
5.1 Overview 92
5.2 Future Work 94

Bibliography 95

List of Figures

2.1 Starting Vectors and Matrices for Attention 10
2.2 Computing Attention for One Token 12
2.3 Transformer Architecture 14
2.4 Input Transformations for Fine-tuning GPT on Different Tasks 18
2.5 Masked Language Modelling 19
2.6 Fine-tuning BERT on Different Tasks 20
2.7 Benefits of Joint Learning 26
2.8 Entity Ambiguity Example 29

3.1 NER Pipeline Part 1 49
3.2 NER Pipeline Part 2 50
3.3 CBOW and Skip-gram Architectures 51
3.4 Self-alignment Pre-training Procedure 59
3.5 EL Training and Test Files 64
3.6 EL Dictionary Files 65
3.7 EL Pipeline Part 1 66
3.8 EL Pipeline Part 2 67

List of Tables

2.1 SympTEMIST's SymptomNER Subtask Results 32

2.2 SympTEMIST's SymptomNorm Subtask Results 33

2.3 SympTEMIST's SymptomMultiNorm Subtask Results 33

2.4 CANTEMIST's CANTEMIST-NER Subtask Results 34

2.5 MultiCardioNER's MultiDrug Subtask Results 35

2.6 MultiCardioNER's CardioDis Subtask Results 36

3.1 Unprocessed SympTEMIST's Entity Mention Statistics 39

3.2 Unprocessed SympTEMIST's Code Statistics 39

3.3 Unprocessed CANTEMIST's Entity Mention Statistics 40

3.4 Unprocessed MultiCardioNER's Entity Mention Statistics 41

3.5 Unprocessed CardioCCC's Entity Mention Statistics 41

3.6 NER Models 44

3.7 NER Hyperparameter Combination 45

3.8 NER SympTEMIST's Statistics 47

3.9 NER CANTEMIST's Statistics 48

3.10 NER MultiCardioNER's Statistics 48

3.11 Training Corpora Sizes for the Word2Vec Models 52

3.12 NER Word2Vec-augmented Training Files' Statistics 55

3.13 NER FastText-augmented Training Files' Statistics 56

3.14 NER Examples of Augmented Sentences 57

3.15 EL Models 61

3.16 EL Hyperparameter Combinations 62

3.17 EL Hyperparameter Search Spaces 63

3.18 EL SympTEMIST's Statistics 64

3.19 EL Dictionary Experiments 65

3.20 EL Word2Vec-augmented Training Files' Statistics 69

3.21 EL FastText-augmented Training Files' Statistics 70

3.22 EL Examples of Augmented Mentions 70

4.1 NER Transfer Learning Results . . . 72
4.2 NER Means, Medians and Best Reported F1-scores . . . 72
4.3 NER Average Entity Mention Length . . . 73
4.4 NER Percentages of False Positives and False Negatives . . . 74
4.5 NER Word2Vec Data Augmentation Results . . . 76
4.6 NER Word2Vec Data Augmentation Best Results . . . 78
4.7 NER FastText Data Augmentation Results . . . 79
4.8 NER FastText Data Augmentation Best Results . . . 80
4.9 NER Best Overall Results . . . 81
4.10 NER Result Comparison . . . 81
4.11 EL Transfer Learning Results . . . 82
4.12 EL Means, Medians and Best Reported Accuracies . . . 83
4.13 EL Dictionary Comparison . . . 84
4.14 EL Fine-tuning Comparison . . . 84
4.15 EL Word2Vec Data Augmentation Results . . . 86
4.16 EL Word2Vec Data Augmentation Best Results . . . 87
4.17 EL FastText Data Augmentation Results . . . 89
4.18 EL FastText Data Augmentation Best Results . . . 90
4.19 EL Best Overall Results . . . 90
4.20 EL Result Comparison . . . 91

List of Listings

3.1 IOB Tagging Scheme Example 42
3.2 ANN File Example 46
3.3 CONLL File Example 47
3.4 NER Data Augmentation Algorithm Pseudocode 54
3.5 EL Data Augmentation Algorithm Pseudocode. 68

Glossary

Corpus

A large, structured collection of texts used for research and analysis. In Natural Language Processing, they are used for training and evaluating approaches like the usage of Large Language Models. *(pp. 2, 7, 16, 19, 23, 26, 28, 32, 34, 35, 38, 40, 43–46, 52, 53, 72, 75, 78)*

Data Augmentation

A technique that increases the diversity of training data (be it text, images, audio, etc) by creating new data samples through transformations or variations of the existing data. *(pp. v, vi, 2–4, 30, 31, 37, 41, 51, 52, 54, 57, 66, 68, 71, 72, 74–82, 85–94)*

Entity

A distinct, identifiable object or concept in the real world, such as a person, organization, location, or in the biomedical domain, a disease, medication, etc. *(pp. 1, 6, 22–29, 31, 34, 38, 40, 42, 43, 45, 46, 48, 53, 58, 72–74, 87, 88, 91, 92, 94)*

Entity Linking

A Natural Language Processing task that associates mention of a given type (e.g. disease, medication, etc) with entries in a knowledge base. *(pp. v, vi, 1–4, 6, 21, 24–29, 32, 37–39, 41, 57, 60, 61, 63, 66, 71, 82, 87, 90–94)*

Entity Mention

A specific reference to an entity within written text. *(pp. 22, 23, 25–30, 32, 34, 39–43, 45–51, 53–60, 63–67, 69, 70, 73, 75, 83, 85, 87, 88, 91–93)*

Knowledge Base

A structured collection of information about facts (entities) and the relationships between them. *(pp. 1, 22, 25–30, 57, 60, 87, 93)*

Large Language Model Large-scale machine learning models that are trained on vast corpora to perform various language tasks. Also known as Foundation Models. *(pp. 3, 4, 16, 21, 24, 28, 51, 72)*

Named Entity Recognition A Natural Language Processing task that identifies and classifies mention of a given type (e.g. disease, medication. etc) within text. *(pp. v, vi, 1–4, 6, 21, 22, 25, 29, 31, 37, 41–46, 48, 57, 66, 71, 75, 80, 81, 87, 90, 92–94)*

Natural Language Processing Subfield of Artificial Intelligence focused on developing algorithms and models that enable machines to understand and generate human language. It includes tasks like machine translation, sentiment analysis, etc. *(pp. v, vi, 1, 4, 6, 8, 16, 22, 30, 37, 38)*

Shared Task A competitive research challenge where participants attempt to solve a specific problem using a standardized dataset and evaluation metric, in order to facilitate the comparison between approaches. *(pp. v, vi, 3, 4, 23, 27, 31, 37–41, 43–46, 49, 50, 60, 71, 72, 92, 93)*

Token A basic unit of text. It could be a word, a punctuation mark, a subword (like "to" or "ken" in "token"), depending on the implementation. *(pp. 8–13, 15, 18–20, 22, 24, 25, 31, 42, 43, 46, 47, 49, 50)*

Transfer Learning A machine learning technique where knowledge gained from one task is applied to improve performance in a different but related task. In the context of Named Entity Recognition and Entity Linking, it involves adapting pre-trained Large Language Model to new datasets (a process called fine-tuning). *(pp. v, vi, 2–4, 16, 17, 30, 31, 37, 41, 57, 71, 72, 74, 75, 77, 80–82, 87, 88, 90–94)*

Acronyms

BiLSTM	Bidirectional Long Short-Term Memory *(p. 24)*
BRAT	Brat Rapid Annotation Tool *(p. 46)*
CBOW	Continuous Bag of Words *(pp. 51, 52)*
CONLL	Conference on Natural Language Learning *(pp. 46, 47, 49, 53, 74)*
CRF	Conditional Random Fields *(pp. 24, 30)*
DNN	Deep Neural Network *(p. 7)*
EL	Entity Linking *(pp. v, vi, 1, 3, 4, 21, 25–29, 32, 34, 37, 38, 40, 52, 57, 61, 63–71, 82, 83, 85–94)*
FN	False Negatives *(pp. 22, 23)*
FP	False Positives *(pp. 22, 23)*
ICD-O-3	International Classification of Diseases for Oncology, 3rd Edition *(p. 34)*
JSON	JavaScript Object Notation *(p. 50)*
LLM	Large Language Model *(pp. 3, 4, 16, 17, 21, 24, 28, 30, 51, 70)*
MEL	Multilingual Entity Linking *(pp. 32, 38)*
MeSH	Medical Subject Headings *(p. 26)*
MLM	Masked Language Modeling *(p. 19)*
MTL	Multi-Task Learning *(p. 94)*
NER	Named Entity Recognition *(pp. v, vi, 1, 3, 4, 21, 22, 24–28, 30, 32, 34, 37, 38, 40, 42–44, 48, 50, 52–57, 66, 67, 71–74, 76–81, 85–87, 90–94)*
NLP	Natural Language Processing *(pp. 1, 2, 6–8, 16, 17, 31, 37, 51)*

NSP	Next Sentence Prediction *(p. 19)*
NSS	Nearest Neighbour Search *(p. 60)*
OMIM	Online Mendelian Inheritance in Man *(p. 26)*
pp	percentage points *(pp. 71, 72, 75, 77, 80, 83–85, 87, 92, 93)*
SAP	Self-Alignment Pre-training *(pp. 58–60)*
SPACCC	Spanish Clinical Case Corpus *(pp. 38, 40, 45)*
TP	True Positives *(pp. 22, 23)*
TSV	Tab Separated Values *(pp. 50, 63, 74)*
UMLS	Unified Medical Language System *(pp. 1, 26, 30, 58, 59, 83)*

1

Introduction

1.1 Motivation

Biomedicine is an ever-growing and complicated domain, evidenced by the immense amount of literature on the subject, with the volume of biomedical material continuing to grow exponentially. Documents like research papers, clinical reports and trials, textbooks and educational materials, and even biomedical newsletters may contain valuable and decisive information not only for research, but for healthcare as a whole. However, this massive reserve of biomedical knowledge is predominantly present in free-text documents, a very loosely structured medium that computers do not understand, which makes it inefficient to extract the information that they contain. Additionally, the complicated terminology of the domain is another challenge as its intricacy hinders automated systems in comprehending nuanced scientific concepts.

Deep learning techniques have surged onto the Natural Language Processing (NLP) landscape in recent years, and present themselves as a powerful option to extract complex information embedded in free-text documents. More specifically, recent Transformer-based models are shown to be able to grasp context and semantic relationships, and thus have pushed the state-of-the-art when it comes to human-language processing (Mahowald et al., 2023). Information extraction-related tasks such as Named Entity Recognition (NER) - the identification of specific entities within text, including genes, proteins or diseases (when referring to the biomedical domain) - and Entity Linking (EL) - the association of those entities with corresponding entries in a knowledge base - represent the course of action to take in order to harvest this information, which can significantly enhance several critical tasks in biomedical research like database curation, hypothesis generation, semantic search, and others.

On the other hand, the vastness of the biomedical domain extends beyond linguistic borders and encompasses a diverse range of languages and cultural contexts. Despite of this, the majority of the resources are allocated to English (Crema et al., 2022), while other languages contain valuable insights that should be easily accessible and not hindered by an idiomatic barrier. An example of this is the popular dataset UMLS (Bodenreider, 2004), where the coverage in English is six times bigger than in Spanish, and twenty-four times bigger than in French (Liu, Vulić, et al., 2021). Knowledge sharing between languages can help bridge this gap and promote the global collaboration of researchers and scientists around the world, which is to be not only incentivised, but made as frictionless as possible in order for scientific advancements to flourish unobstructed and to propel the pace of discovery and innovation. Furthermore, bridging this multilingual gap is also vital in the sense that medical information should be accessible by all healthcare practitioners in order to provide the best possible care to those who seek it. This motivates the search for solutions that are increasingly more reliable in a multilingual context, as their performance still pales in comparison to the state-of-the-art performance on English texts (French & McInnes, 2023).

Transformer-based models support cross-lingual transfer-learning, which means that they can be trained on large corpora in one language, and then be adapted to another language where the amount of data is limited. Research has shown that non-english NLP systems in low-resource scenarios can benefit from pre- trained english language models (Chi et al., 2024). Additionally, this transfer of knowledge can also happen between non-English languages (Andrade et al., 2021). Characteristics like these make Transformer-based approaches and techniques well-suited candidates for tackling the challenges of extracting information present in the intricate language structures of biomedical literature, not only in one language but in a multitude of them.

1.2 Objectives

This dissertation provides a comprehensive exploration of some recent Transformer-based models and techniques for Named Entity Recognition and Entity Linking within the biomedical domain. Its primary objective is to extend the application of these state-of-the-art models to a multilingual scenario by employing multiple approaches and datasets, thereby contributing to the discussion and advancement of biomedical information extraction in the multilingual paradigm.

The first approach was to look at the efficacy of transfer learning approaches when adapting Transformer-based models like BERT and its derivations to the complexities of multilingual biological texts. More specifically, the approach of further training and consequently tweaking (fine-tuning) a model that has already been trained on large and open-domain datasets, but now for target languages on less abundant domain-specific data, was investigated in order to verify the hypothesis that it can improve the models' ability to recognise and link concepts across linguistic borders.

Next, data augmentation techniques were used to try and mitigate the relative scarcity of labelled data in the biological domain, of languages other than English. Techniques like synonym replacement seek to enrich the training data and, by extension, boost Transformer-based models' performance against the challenges posed by diverse linguistic expressions and variations within the biomedical domain across languages.

There exists a series of shared tasks available with the intent of evaluating state-of-the-art information extraction models, mainly through NER and EL, for the biomedical domain. They take submissions from researchers all around the world and rank them based on their performance on a specific objective and dataset. We chose the following to evaluate our approaches on: SympTEMIST (Lima-López, Farré-Maduell, Gasco-Sánchez, et al., 2023), which targets the detection and normalization of symptoms, signs, and findings, primarily in Spanish but also includes an experimental subtask for other languages using SNOMED CT[1]; CANTEMIST (Miranda-Escalada et al., 2020), which involves the identification and normalization of tumor morphology concepts in Spanish documents; and MultiCardioNER (Lima-López et al., 2024), which addresses the multilingual adaptation of clinical NER systems for cardiology, focusing on diseases and medications across different languages. The approaches explored in this dissertation were evaluated in the context of these shared tasks, in order to obtain a solid comparison against the current state-of-the-art.

In short, this work aims to explore transfer learning and data augmentation techniques for the tasks of Named Entity Recognition and Entity Linking, in the biomedical domain and on a multilingual level.

1.3 Contributions

The research presented in this dissertation makes a few contributions to the field of biomedical information extraction, particularly in the context of multilingual Named Entity Recognition and Entity Linking:

- Application and comparative analysis of various Large Language Models (LLMs) to perform NER and EL on the SympTEMIST, CANTEMIST and MultiCardioNER shared tasks. The results achieved were highly competitive, with many results surpassing the mean and median scores for their respective tracks, and one notable NER result exceeding the best reported F1-score for its respective track - DrugTEMIST English.

- A participation in the MultiCardioNER shared task, along with a complementary system description paper (Gonçalves & Lamúrias, 2024), published in CLEF's 2024 Working Notes (Faggioli et al., 2024), that outlines the employed approach[2].

- Evaluation of 13 LLMs in the context of SympTEMIST's EL subtasks, and 1 in the context of MultiCardioNER, marking their first application in these tasks and providing new evaluation scores and results that contribute to the benchmarking of these models in multilingual biomedical information extraction.

[1] https://www.snomed.org/five-step-briefing

[2] Although the official results reported by the competition regarding our submission seem poor, this was due to a submission error on our part, with the corrected (albeit local) evaluation indicating a significantly better and very competitive performance.

- Implementation of a synonym replacement data augmentation technique to enrich both NER and EL training data, using open-source pre-trained FastText word embeddings (Bojanowski et al., 2017) from various languages (Grave et al., 2018), as well as training our own Word2Vec word embedding models (Mikolov et al., 2013) for those same languages.
- Public release of the codebase[3] used to conduct all experiments. This includes the dataset preprocessing scripts, the LLM fine-tuning scripts, and the data augmentation scripts to train the Word2Vec models and augment the datasets, ensuring that future researchers can replicate and build upon the findings of this research.

The contributions outlined above collectively offer new insights for future research on Transformer-based approaches for multilingual biomedical text processing. With these contributions, we were able to develop approaches that yielded competitive scores and an overall noteworthy performance on the tasks we set out to participate in. Furthermore, we believe that our final analysis poses valuable comparisons and insights that can guide future research and applications in multilingual biomedical information extraction. We also expect that the approaches developed in this dissertation can be applied to other shared tasks, challenges and datasets within the biomedical domain, with support for any language (provided there is a Large Language Model for that language).

1.4 Document Structure

In addition to the current chapter, this document is composed of four additional ones:

- Chapter 2 explores the necessary concepts in order to understand the work carried out in this dissertation. It begins by providing an introduction on Natural Language Processing, before narrowing the focus to the Attention mechanism and introducing Large Language Models. This is followed by a description of the two major tasks addressed in this dissertation - Named Entity Recognition and Entity Linking - then the strategies used to tackle those tasks, and lastly how the efficacy of those strategies is assessed.
- Chapter 3 outlines the experiments that were conducted in this dissertation. It begins with a quick overview of some of the computational tools required to carry out this research, followed by a full description of the datasets that were used, before finally covering every stage of the design and implementation of the transfer learning and data augmentation techniques, for both NER and EL.
- Chapter 4 presents the results of our NER and EL approaches, offering a discussion of the findings and interpreting them not only in the context of the shared tasks

[3] https://github.com/Rodrigo1771/multilingual-bio-ner-and-el-msc-diss

themselves, but in the context of the multilingual biomedical domain as a whole, also touching on the limitations of the approaches and challenges encountered.

- Chapter 5 concludes the dissertation by summarizing the key findings and contributions the research. It addresses any remaining questions and considers future research directions that could further improve the developed approaches.

2

Related Work

2.1 Natural Language Processing

2.1.1 Definition

Natural Language Processing (NLP), refers to the branch of Artificial Intelligence that gives machines the ability to read, understand, and derive meaning from human language. It aims to produce models that can break down and separate significant details and contextual nuances from regular text and speech, just like or close to what humans can. This can then be applied to a wide range of tasks, such as machine translation and automatic summarization, text-to-image or text-to-video generation, or the ones which are the focus of this dissertation and will be covered in depth, Named Entity Recognition and Entity Linking.

2.1.2 Brief History

Throughout history, Natural Language Processing has given rise to a multitude of techniques depending on the task at hand (Johri et al., 2021; Louis, 2020a, 2020b). For example, one of the very first applications of NLP was in the area of machine translation, when an automatic text translation program was developed through a joint project between Georgetown University and IBM Company in 1954.

Advancements in the subsequent decades all fell in a general category that can be denominated symbolic approaches (Charniak, 1983; C. E. Martin & Riesbeck, 1986). These types of approaches involved representing information as symbols, that is, abstract representations that carry the meaning of knowledge components like entities or relationships, and operations, also referred as (hand-written) rules, that were performed to those symbols to derive conclusions and make decisions.

It wasn't until the 1980s that another kind of approach started to gain some traction: statistical models. These techniques often involve making probabilistic inferences about linguistic structures or sequences, based on the statistical properties learned during the training phase. Techniques like decision trees (Allmuallim et al., 1994; Quinlan, 1986), which use if-then rules to better deduce the optimal result, and word n-grams (Brown et al., 1992), which assume that the probability of the next word in a sequence depends on a fixed number of previous words, were only possible because of the advancements in computational power and the shift to machine learning algorithms, reliant on large amounts of data, which stripped away the complexity of having to write and tweak hand written rules while offering initially similar performances.

With the turn of the century, we started to see yet another different kind of approach: neural models. A crucial event responsible for making this switch was when a multi-layer perceptron surpassed the word-n gram, the best performing statistical model at the time (Bengio et al., 2003). This marked a turning point in the NLP space as they also introduced a concept known as word embedding. This is a technique that attempts to best represent words as vectors in a continuous vector space based on their meaning and usage in text. The technique would then be improved into, among others, the widely adopted word embedding model: Word2Vec (Mikolov et al., 2013). With this embedding model, the researchers were able to improve on the efficiency and accuracy of the training procedure, which opened the doors for larger and larger training corpora to be used for training. But the major reason that this and other embedding models are so relevant nowadays is the proven performance improvement in various tasks when pre-trained word embeddings are used. This majorly improves the speed and complexity of research since training these embeddings is slow and costly, and employing these pre-trained embeddings means that it is not generally necessary to learn the embeddings from scratch, instead relying on pre-trained ones.

Finally, the machine learning paradigm shifted once more, and Deep Learning quickly became the most adopted approach in the NLP space. Because Deep Learning can extract sophisticated patterns and representations from large, complicated datasets, it has attracted a lot of interest over the past few years and has grown to be a game changer in this field. Deep learning approaches employ Deep Neural Networks (DNNs), i.e., neural networks with several hidden layers. This type of network, being composed by many computational units working in tandem, is inspired by the complex processes of the human brain that enable information processing and self-learning. Convolutional Neural Networks (Lecun et al., 1998) and Recurrent Neural Networks (Rumelhart et al., 1986) (the latter steadily replaced by Long Short-Term Memory networks, Hochreiter and Schmidhuber, 1997) were initially some of the most popular DNN architectures. However, an important leap forward and a precursor to the architecture and techniques that are fundamental to this dissertation - namely the attention mechanism (Bahdanau et al., 2016) and the transformer architecture (Vaswani et al., 2023) - was the development of sequence-to-sequence learning (Sutskever et al., 2014). The model was composed of an encoder that processes the input sequence and produces a vectorized representation of it, and a decoder that iteratively tries to predict the output sequence taking the encoder's representation along with everything it predicted previously. This technique is very suited for tasks like machine translation or text summarization. While it did not necessarily introduce the concept of an encoder and a decoder, it was one of the first to do so, and it was a major step towards the current landscape of Deep Learning in Natural Language Processing.

2.2 Attention Mechanism and Transformers

As mentioned in the previous section, the attention mechanism (Bahdanau et al., 2016) was initially intended to be an improvement on the encoder-decoder model in sequence-to-sequence learning (Sutskever et al., 2014). However, with the proposal of the transformer architecture (Vaswani et al., 2023), attention quickly jumped to the frontier of innovation and has made it into most state-of-the-art systems for NLP today, especially with the current framework of pre-training models on huge amounts of generic data and then fine-tuning them for a specific task.

2.2.1 Input Preprocessing Steps

Before discussing the attention mechanism, it is important to know how these sequences are prepared in order to be fed as input to the transformer model.

2.2.1.1 Tokenization and Word Embedding

Firstly, the input sequence is tokenized, i.e. it is broken down into smaller units, known as tokens. These tokens can vary from whole words to smaller subwords (for simplicity's sake, a token will be treated as a word in this explanation). Each of these tokens is then mapped to a vector of real numbers, also referred to as an embedding vector. These vectors, of dimensionality (size) d_{emb}, are able to capture the semantic information of each token. As such, the embedding process involves looking up the pre-trained embedding vector for each token in the sequence, in a pre-defined embeddings matrix. This matrix, of dimensions $n_{\text{tokens}} \times d_{\text{emb}}$ (with n_{tokens} being the vocabulary of the model, i.e., every token that the model knows), is part of the model's parameters, and is randomly initialized and learned during the training process.

2.2.1.2 Positional Encoding

One problem with these embeddings is that they do not encode positional information, a very useful feature. In other words, with just these embeddings, the attention mechanism cannot infer the position of each token in the input sequence. The way this information is added is by computing position embedding vectors and adding them to the embeddings obtained in the previous step. Unlike the previous embedding vectors, these ones are not learned, instead they are only computed once and reused for every sentence. This means that, for different input sequences, the positional encoding vectors for tokens in different sequences will be equal if their position inside their respective sequence is also equal. The expressions to calculate these positional encoding vectors are the following:

$$\mathrm{PE}_{(pos,2i)} = \sin\left(\frac{\mathrm{pos}}{10000^{\frac{2i}{d_{\mathrm{emb}}}}}\right)$$

$$\mathrm{PE}_{(pos,2i+1)} = \cos\left(\frac{\mathrm{pos}}{10000^{\frac{2i}{d_{\mathrm{emb}}}}}\right)$$

Both expressions take two arguments: first, the position of the token inside the input sequence, and then the dimension (embedding vector index) for which the value is being calculated. The first expression is used if that dimension is even, and the second one if it is odd. After doing this for each dimension of every token in the input sequence, we are left with these positional encoding vectors that are added to the original embeddings to form the input for the model.

2.2.2 Attention

Now that the input sequence is represented in a way that the model can understand, we can move on to the attention mechanism.

2.2.2.1 Definition and Motivation

Attention is a technique that allows neural networks to selectively focus on specific parts of the input data, rather than treating all parts equally. It enables the network to learn which parts of the input data are most relevant to the task at hand. More specifically, self-attention (the one commonly used) does this by relating words to each other in the same input sequence. To understand why this is useful, let's say we have the following sentence:

The *dog* sleeps to *its* own bed.

Obviously, we know that *it* refers to *dog* and not anything else, like *bed*. But can the model understand this? Through attention, the model has the ability to figure out that *it* refers to *dog*, by outputting representations that resemble each other for similar words.

Furthermore, consider the following two sentences:

The *bat* flew through the dark cave.
The baseball player prepared to swing his *bat* at the incoming pitch.

Again, it is obvious to us that the word *bat* has two different meanings, as it is used in a different context in each of the sentences. However, how does the model know this? By having different representations for both words based on the context surrounding them, i.e. the other words in the sentence, it knows that *bat* means different things, depending on the sentence.

With these examples its clear that if this is scaled to encompass everything from synonyms, to plurality, to grammar, and even to such an abstract concept like context, the model can focus on the most informative parts of the input, ignoring irrelevant or redundant information. Thus, the main benefit of attention is that by employing this mechanism the network is able to learn the overall structure of the language that it is being fed, and the better it learns it, the better it performs on any number of language tasks.

2.2.2.2 Single-Head Self-Attention

Self-attention is the technique of applying the attention mechanism between every two tokens of a sequence, essentially relating the sequence to itself. The prefix "single-head" indicates that this process is computed once, as opposed to multi-head self-attention which is computed multiple times and will be explained shortly after. Single-head self-attention can be decomposed in 3 basic steps:

1. Obtaining the queries, keys, and values for each token.
2. Calculate the attention weights (with the attention scores).
3. Obtain the output vector by adding the resulting weights to each corresponding value.

The first step is to obtain the **q**uery, **k**ey and **v**alue vectors of each token. The **q**uery and **k**ey vectors have a dimension (length) of d_k, while the **v**alue vector has a dimension of d_v. These vectors are concatenated and treated as matrices to help with computation efficiency, leading to the **Q**ueries, **K**eys and **V**alues matrices (Figure 2.1). These matrices (and consequently vectors) are basically projections of the original input, obtained by 1) concatenating the embeddings from the input sequence into a matrix, and then 2) multiplying that matrix by each of three projection matrices, each one randomly initialized and learned during the training process.

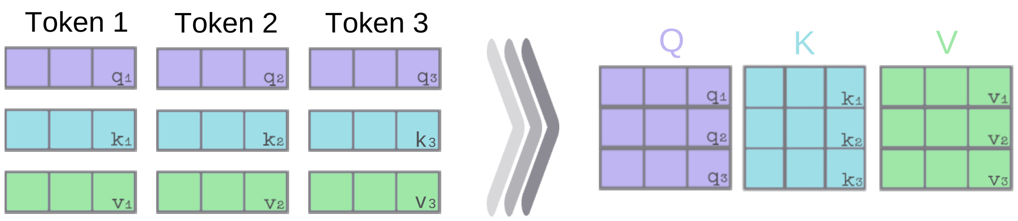


Figure 2.1: Each token will produce a **q**uery, **k**ey and **v**alue vector, which together will form the **Q**ueries, **K**eys and **V**alues matrices. For simplicity, their true dimensions d_k and d_v are not represented.

The second step is to calculate the attention weights. To do this, we first need to obtain the attention scores, which are provided by the scoring function. This function is used to relate each token to every single token in the input sequence. It represents the amount of emphasis the model needs to put into every token in the sequence, when encoding a given token. There are some scoring functions to choose from, although the most popular one and the one used in the original paper (Vaswani et al., 2023) is the scaled dot product scoring function. This function computes the matrix multiplication between the **Q**ueries and **K**eys matrices, which is essentially the dot product of each **q**uery vector with each and every **k**ey vector. However, in the case of large vector representations, those dot products end up being very large, which is a problem: very large dot products mean very small gradients during backpropagation, which can make it challenging for the model to learn effectively. To combat this issue, the resulting matrix is divided by the square root of the dimension of the vectors, d_k (in practice, this equates to dividing each dot product by d_k). In other words, it is scaled down (hence the name) by this value. Thus, the scoring function can be written as

$$\frac{QK^T}{\sqrt{d_k}}.$$

The attention weights are then obtained by normalizing these attention scores, ensuring that they all sum up to one. This normalization process is usually done through the use of the *softmax* operation, as it is a very popular choice for obtaining a probability distribution given a set of values (with x being that set of values, and i and j the index corresponding to a given value):

$$\mathrm{softmax}(\mathbf{x})_i = \frac{\exp(x_i)}{\sum_j \exp(x_j)}$$

Putting it all together, the whole computation can be written as

$$\mathrm{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right).$$

The last step envolves the **V**alues matrix calculated earlier. When calculating attention for a given token (despite it not being calculated in such an iterative way, but it simplifies the explanation), the **V**alues matrix is multiplied by the weights from the last step (in other words, each **v**alue vector is multiplied by the corresponding weight). Because the weight is a scalar, the resulting matrix is of same height and width. Lastly, the vectors that make up that resulting matrix are summed to obtain a single vector. Basically, all that is being done is a weighted sum of (the **v**alue vector of) each token by its significance (the actual weight) to the token we are obtaining the attention of. The intuition here is to keep intact the values of the most relevant tokens given a token of the input sequence, i.e. the tokens that the network should focus on when operating on that token, as these will have greater weights. Likewise, the network will likely ignore the irrelevant ones, as these will have very small weights. Each dimension of this output vector represents a token in the input sequence, and each value within that dimension represents how much that token is related to the token that attention was calculated for. This computation is done for every single token through matrix calculations, thus resulting in a final matrix, i.e. one of these output vectors for each input token. The final expression for attention can be written as

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

In the end, we will end up with an output matrix, where each vector corresponds to a token in the input sequence. Figure 2.2 can help to digest the whole mechanism:

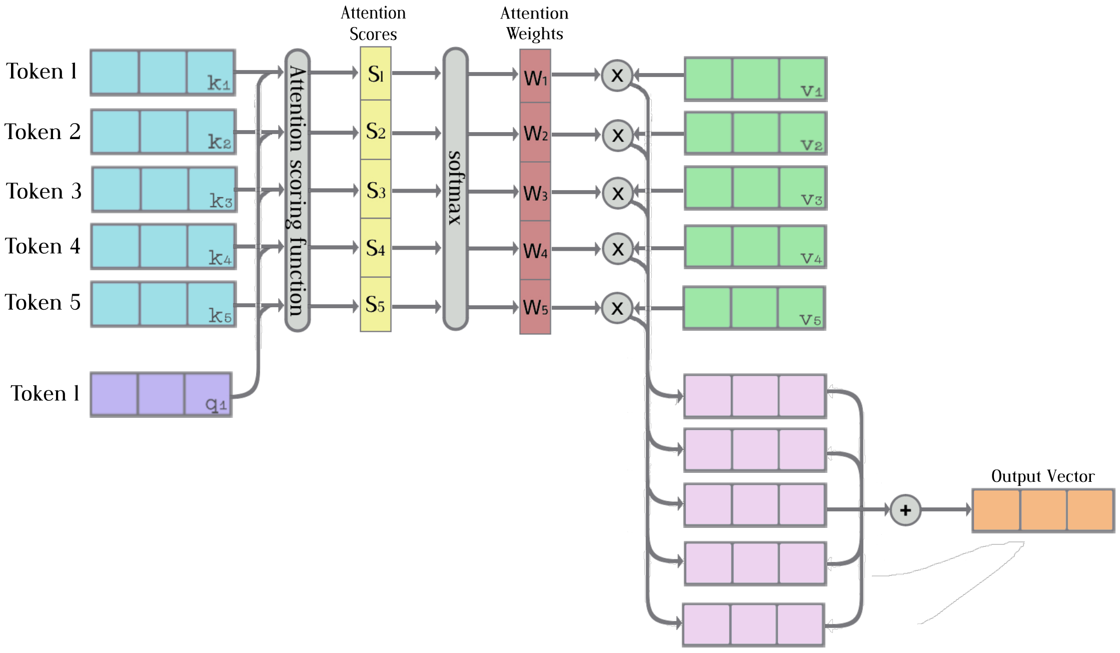


Figure 2.2: Computing attention for "Token 1". The attention scores are computed by passing both the **q**uery vector of the token in question, and the **k**ey vectors from every token in the input sequence (i.e. the **K**eys matrix), to the attention scoring function, which in this case is the scaled dot product scoring function. Then, those attention scores are passed through the *softmax* function in order to obtain the attention weights. Each one of these is multiplied by the **v**alue vector of the corresponding token, and finally everything is summed to obtain a single vector. For simplicity, dimensions are not to scale. Inspired by Zhang et al. (2023).

This is the basic idea behind self-attention. However, the transformer architecture employs a variation of self-attention, namely multi-head attention layers. It was an refinement to the original mechanism, developed in the same paper, to improve the performance of the attention layers.

2.2.2.3 Multi-Head Self-Attention

Multi-head self-attention is essentially self-attention performed multiple times in parallel, by different heads. What this means is that the **Q**ueries, **K**eys and **V**alues matrices are linearly projected h times (the number of heads), in order to obtain different representations of the input sequence. This helps the model to focus on different positions. Because each head has its own projection matrices, initialised randomly and learnable through training just as previously stated, each one will give rise to different **Q**ueries, **K**eys and **V**alues matrices. Thus, the attention mechanism performed by each head will output a different vector representation for each token in the end, or in other words, a different matrix representation for the input sequence.

However, after performing attention h times, we are left with h matrices, each one of them corresponding to a representation of the input sequence, which is not the desired output as the model's architecture (discussed below) requires a single matrix. This is the final step to multi-head self-attention, where we concatenate these h matrices, take the resulting structure and multiply it by another matrix that is randomly initialized and learned through training, just like the weight matrices for obtaining the **Q**ueries, **K**eys and **V**alues matrices. This results in a single matrix that gathers various representations of the same word, each one attending to different parts of the input.

That concludes the concepts behind the mechanism that gives the transformer its power, while its architecture will be explored next.

2.2.3 The Transformer Architecture

The transformer (Vaswani et al., 2023) consists of two main components: the encoder and the decoder. Their compositions are somewhat similar, but have some key differences according to their function in the overall architecture. First, both are stacks of a number N (6 in the original paper) of identical layers. Furthermore, a residual connection is used alongside each of the sublayers that make up a layer, along with layer normalization at the end of that same sublayer. Thus, the output of a given sublayer will be the normalized sum between its input and output. The purpose of this residual connection is to facilitate gradient flow and, again, mitigate the vanishing gradient problem.

Before focusing on each of the two layers that make up the foundation for both the encoder and the decoder stacks, it is worth noting that "attention" and "self-attention" are used interchangeably throught literature, and likewise in the following sections, when referring to self-attention. Now, a single encoder layer is structured in the following way:

- A multi-head self-attention layer followed by layer normalization with a residual connection.
- A fully connected feed-forward layer followed by layer normalization with a residual connection.

On the other hand, a single decoder layer is structured in the following way:

- A masked multi-head self-attention layer followed by layer normalization with a residual connection.

- A multi-head self-attention layer that performs multi-head self-attention aided by the output of the encoder stack (this is also known as encoder-decoder attention or cross-attention), followed by layer normalization with a residual connection.
- A fully connected feed-forward layer followed by layer normalization with a residual connection.

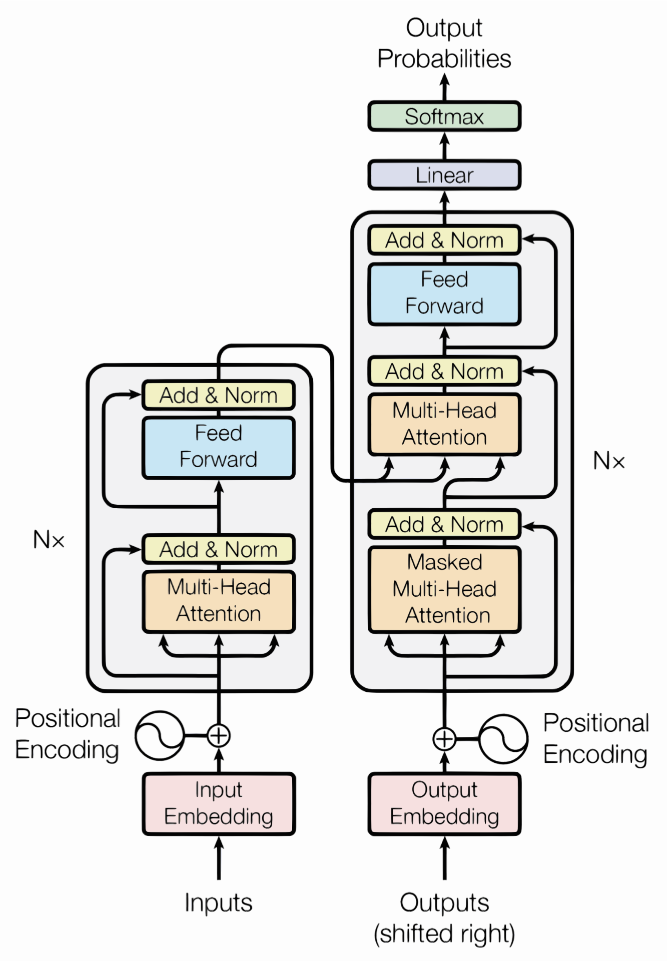


Figure 2.3: The Transformer architecture. Adapted from Vaswani et al., 2023.

2.2.3.1 Encoder

The inner workings of an encoder layer, namely the multi-head self-attention layer, have already been explored in detail. Therefore, all that is left is to emphasise that the encoder is a set of encoder layers stacked on top of each other. This means that the output of one encoder layer will be the input of the next, until it reaches the top encoder layer. The final output of the whole encoder component will be a set of context vectors, where each context vector corresponds to a position in the input sequence. These, however, will not be used. Instead, the **K**eys and a **V**alues matrices from the last encoder layer will be used as input in the cross-attention sublayer of each decoder layer.

2.2.3.2 Decoder

The first sublayer of a decoder layer is the masked multi-head attention sublayer. This sublayer is basically a multi-head attention sublayer with the added feature that the decoder can only attend to past positions of the input sequence, instead of every position. This is because a decoder generates the output sequence one token at a time, and the intuition behind this first sublayer is that, as the model is generating a token, it can only have the context of the tokens that it has generated before, not the ones that it is still to generate. This is done by masking future positions, i.e. setting them to *-inf* before the *softmax* step in the self-attention calculation, as when *-inf* is fed into the *softmax* operation, the output will be 0, and that token will not have any influence on the self-attention operation.

Right after the masked multi-head attention sublayer is the cross-attention sublayer. As stated before, it is a simple multi-head attention sublayer, with the slight adaptation that it takes the **K**eys and **V**alues matrices from the output of the encoder stack, only needing to calculate the **Q**ueries matrix from output of the previous sublayer. Cross-attention allows the model to attend to different parts of the input sequence (encoder) when generating each element of the output sequence (decoder).

Lastly, for the sake of completing the single decoder layer structure, the last sublayer is a simple fully connected feed-forward network as stated before.

With the main structure now covered, we can proceed to explore how the decoder works in practice, with the help of the encoder, to generate the output.

2.2.3.3 Predicting a Word

The decoder generates the output one token at a time, based on what it has already generated before. So, the first step of the decoder stack at each iteration is to take as input of its first decoder layer the vector embeddings and positional encodings of the output sequence of the previous iteration. In the case of the first iteration, the decoder takes as input a special character that represents the start of the sequence, `<sos>`. The subsequent decoder layers of the stack will take as input the output of the previous decoder layer. Furthermore, inside each decoder layer, the input will first be processed by the masked multi-head self-attention sublayer and the output of that will be passed to the subsequent sublayer, the cross-attention layer. This sublayer will also receive the **K**eys and **V**alues matrices from the output of the encoder stack, and output a vector representation that, after it passes through the fully connected feed-forward network, will be sent to the next decoder layer. This process bubbles up the decoder stack until it reaches the final decoder layer, where a vector of floats is outputted. This will then be converted into a word through the final linear and *softmax* layers. This process repeats itself until the last iteration, where the transformer outputs a special character signaling the end of the sequence, `<eos>`.

2.2.3.4 The Final Linear and Softmax Layer

These two layers convert the output of the decoder stack into a word, by first converting that vector of floats into a logits vector through the linear layer. This is a very large vector in which each position corresponds to a word in the model's vocabulary. In addition, each position has a raw score, that after passing through the *softmax* layer for normalization, represents the probability of it being the right word. The word with the highest probability is chosen, and that is the final step of the transformer architecture.

2.3 Large Language Models

Large Language Models (LLMs) have raised the bar for NLP's transformational potential. Building on the fundamental ideas of the attention mechanism and the transformer architecture (Vaswani et al., 2023), recent research has shown that Natural Language Processing has undergone a paradigm shift. Vast computational resources, combined with creative architectural design and the immense volume of textual data have resulted in Large Language Models, which allow computers to understand and generate human-like text with unprecedented accuracy and fluency.

Large Language Models are a kind of model that can generate human-like text and perform various NLP tasks. They mainly leverage the transformer architecture and its innovations along with truly enormous amounts of data to achieve state-of-the-art performance on those tasks. However, the use of "Large" in Large Language Model is not only due to the gargantuan size of the training data, although the use of such dense corpora on an unsupervised learning framework is a key innovation of these types of models. Additionally, it refers to the size of the model itself, more specifically the parameter count. LLMs typically boast an extensive number of them, typically in the order of hundreds of millions or even billions when referring to the latest state-of-the-art models. This abundance of parameters can be attributed to the scalability of the transformer architecture, that allows these models to take gargantuan proportions, impressive even when taking into account the computational power available.

2.3.1 Pre-Training and Fine-Tuning

A major characteristic of LLMs is that they are general-purpose language models. This is because they are designed with a technique known as Transfer Learning in mind. Transfer Learning stands for the technique where a model trained on one task is used, with or without adjustments, for a second related task. The reuse of the knowledge gained in the first task is intended to boost performance on the second one. In the case of LLMs, this knowledge transfer can happen just via the mechanism called pre-training, or with an additional step named fine-tuning.

First, the idea behind pre-training is to train these models on one or more tasks (examples of such tasks in the sections regarding GPT and BERT), and on huge amounts of generic data, in order for them to learn a general language understanding. This is enough for the model to be able to reuse (transfer) what it learned in training to solve everyday common language problems, like document summarization, question answering and text generation. Additionally, one may wish to apply such a powerful model to a particular domain, and/or to a particular but related task. The problem here lies in the fact that domain-specific datasets are usually many times smaller than the ones used for pre-training, as its often difficult to curate a large quantity of domain-specific data, and so pre-training them with these limited datasets may not yield favourable results. However, LLMs have shown time and time again that a dataset of the magnitude of the ones tipically used in the pre-training process is not needed to achieve state-of-the-art performance, if the second step in the transfer learning process, fine-tuning, is applied.

In fine-tuning, the model is tweaked and refined for the domain/task at hand by further training it on the available domain or task-specific data, resulting in a model that is specifically tailored for tasks that it might not have performed well enough in before this procedure was employed, such as clinical text analysis, code documentation or legal document understanding.

In short, Transfer Learning is the process of leveraging knowledge gained from solving a task, to solve a different but related task, which can be done by either applying a pre-trained model to a general language task, or fine-tuning it to a domain-specific one. Thus, LLMs can be viewed as a sort of general pocket knife for NLP that can be adapted to a bunch of different tasks, also being nicknamed Foundation Models, as they serve as the starting point for those tasks.

2.3.2 Types of Large Language Models

In the previous section (Attention Mechanism and Transformers), we touched on everything about how the transformer architecture enables a model to learn the context behind simple unstructured text. This is done by two major components: the encoder stack and the decoder stack. However, later models developed for NLP do not necessarily follow this structure rigorously, instead applying some of its principals, and innovating on other areas in their own way. For example, while a model like Facebook's BART (Lewis et al., 2019) uses an encoder-decoder model similar to the one previously described, OpenAI's GPT (Radford et al., 2018) uses a decoder only architecture, and Google's BERT (Devlin et al., 2019) an encoder only one. These last two models are the most widely used, and have a few key differences between them that are worth exploring in order to understand some of the major ideas and innovations of the Foundation Models of today.

2.3.2.1 GPT

Starting with OpenAI's GPT (Generative Pre-trained Transformer, Radford et al., 2018), as it was just stated, the model is only made up of a decoder stack. This is due to the pre-training objective of the model, which is just to predict the next word of the sequence based on the preceding words, also known as Autoregressive Language Modelling. The decoder stack is very well suited for this task, as it outputs the words one at a time based on the last ones it predicted. However, since GPT does not use the encoder stack, the decoder layers were modified to exclude the encoder-decoder sublayer, and only the masked multi-head attention and the feed-forward fully connected layers remain. The pre-training process comprises of 1) setting a context window of fixed size; 2) continuously feed the text data to the model; 3) at each time step, letting the model predict the next token based on the tokens present in the context window; and 4) adjusting the weights along the way based on the correctness of those predictions.

As for the fine-tuning process however, a problem arises, as many downstream tasks make use of structured input like ordered sentence pairs, or triplets of document, question, and answers, and GPT's pre-training uses contiguous sequences of text. This is addressed by manipulating the model's input in such a way that the model can process it: concatenating the data with the use of start, end (extract) and delimiter tokens. Then, a linear+*softmax* layer is employed that takes the output of the transformer model (sequence representations) as input and presents the final output. This way, extensive changes to the models architecture in order to cater for a specific task are avoided.

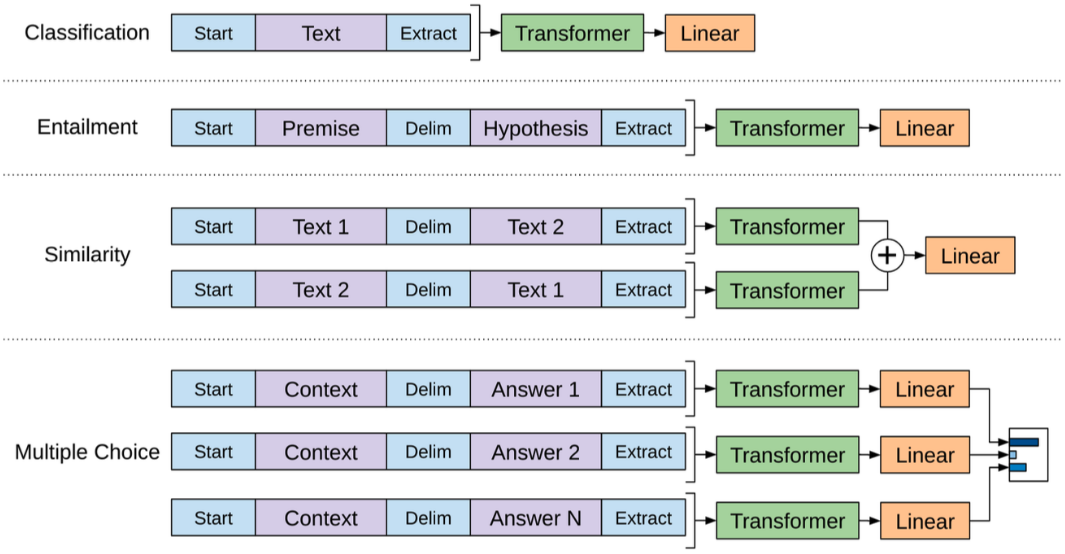


Figure 2.4: Input transformations for fine-tuning GPT on different tasks. For instance, in textual entailment tasks, the premise and the hypothesis are concatenated with a delimiter token in between, while in sentence similarity tasks, two inputs are produced by changing the order of the two sequences when concatenating them, in order to reflect the lack of inherent ordering when comparing two sentences. Adapted from Radford et al., 2018.

This fine-tuning method is called few-shot learning, as the model only needs a few specific examples (in this case given as structured prompts) to achieve a good performance in the downstream task.

2.3.2.2 BERT

Moving on to BERT (Bidirectional Encoder Representations from Transformers, Devlin et al., 2019), the big difference from GPT is that it uses an encoder stack instead of a decoder stack. The purpose of the encoder is to translate tokens into a semantically meaningful representation taking the context of the whole sequence into account. This goal aligns with BERT's focus on bidirectional (hence the name) context modelling, put into practice by its main pre-training objective (in addition to Next Sentence Prediction[1]): Masked Language Modeling (MLM). This pre-training method enables BERT to learn the context of a word not only from the previous words but from the subsequent words as well, by masking or omitting a percentage of the words in the corpus (15% in the original paper), and having the model predict what word fits best with the surrounding context. This way, BERT's pre-training process helps it to better understand context, dependencies and the overall language.

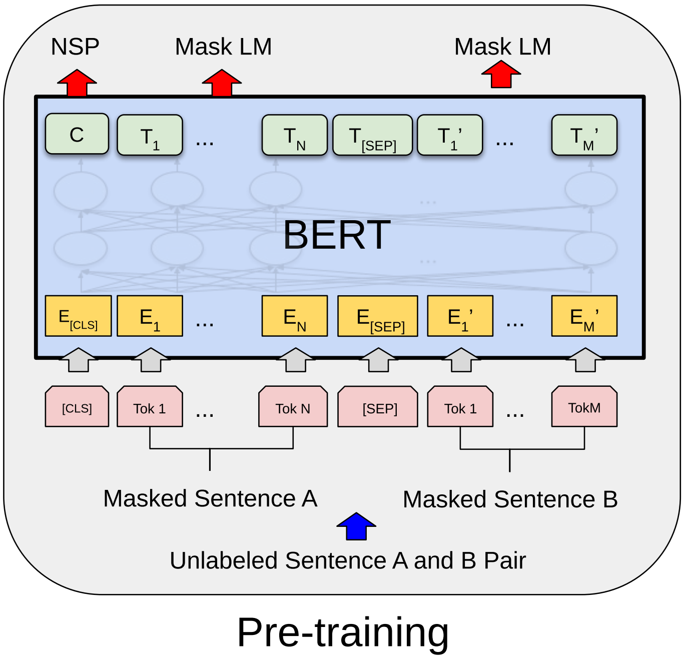


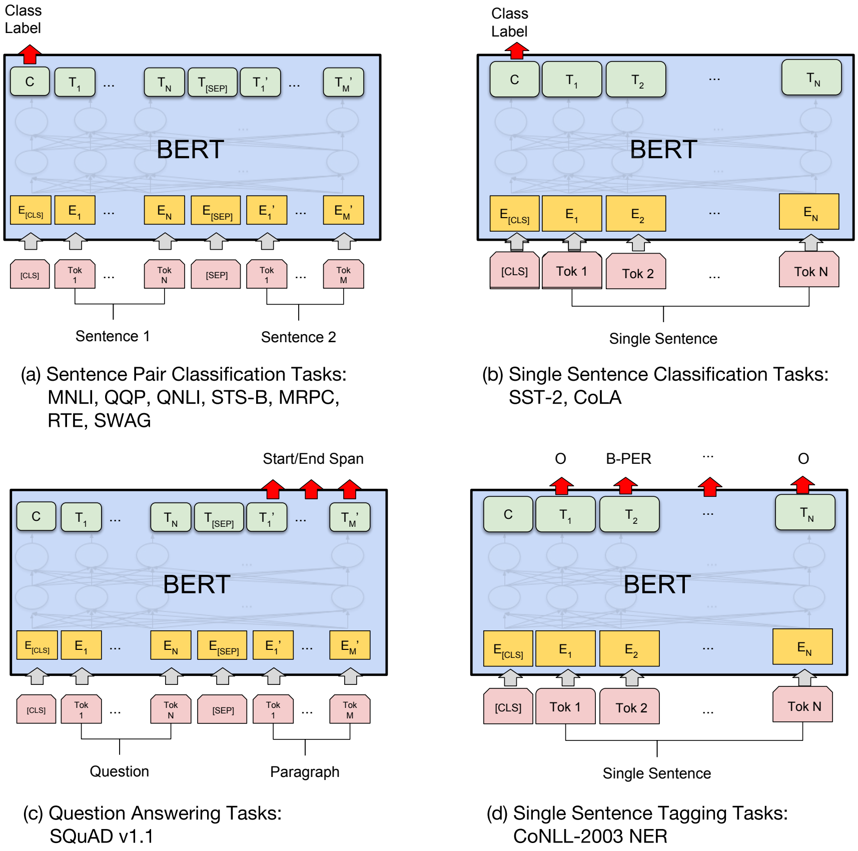
Figure 2.5: BERT's primary pre-training objective: Masked Language Modeling (MLM). The output vector present in the masked word position is BERT's prediction, and will be used to iteratively train the model. Adapted from Devlin et al., 2019.

As explained in the last section (Attention Mechanism and Transformers), the output of the encoder stack is a series of vector representations, one for each token in the input. In the same way, and illustrated in Figure 2.5, when given two sentences with some of its tokens masked, and concatenated using a special separation token `[SEP]`, BERT will also output those vector representations, for each token, masked or unmasked. However, a final

[1] BERT employed another pre-training objective, Next Sentence Prediction (NSP), that aims to predict if a given sentence is likely to follow another. The special classification token `[CLS]` is used as the binary output for this task. However, RoBERTa (a BERT improvement) dropped it as it found that it did not improve the model.

emphasis is put on the representations of the masked tokens, where the loss function that is being minimized (distance between the predicted and the actual word) only accounts for those that are masked in order to force the model to get them right. Those are the basics for BERT's pre-training process.

Transitioning to the fine-tuning phase, BERT makes it simple to adapt the model to the desired task. Depending on the nature of the task, the input can be fed one sentence at a time, like (b) and (d) of Figure 2.6, or two at a time, like (a) and (c). Then, to get the desired output, an additional output layer can be appended to the top of the encoder stack in order to filter for what is needed. For example, in a sentence classification task, one might be interested in the classification output and so a classification layer can be used, but in a question answering task one might only be interested on the span of tokens which correspond to the answer and so an output layer for token-level tasks can be used. This fine-tuning method allows the number of parameters that have to be learned from scratch to be kept at a minimum, as the developer mainly has to train the output layer, with minimal changes happening to the BERT model itself.



(a) Sentence Pair Classification Tasks: MNLI, QQP, QNLI, STS-B, MRPC, RTE, SWAG

(b) Single Sentence Classification Tasks: SST-2, CoLA

(c) Question Answering Tasks: SQuAD v1.1

(d) Single Sentence Tagging Tasks: CoNLL-2003 NER

Figure 2.6: Illustrations of fine-tuning BERT on different tasks. Adapted from Devlin et al., 2019.

2.3.2.3 Domain Specific and Multilingual Models

In the subsequent years to these model's releases, they (among other less adopted models) have been forked and adapted to the biomedical field. Some of these models include but are not limited to BioMedLM (previously PubMedGPT, a clinical adaptation of GPT through fine-tuning it on PubMed's biomedical literature database) on the GPT side, and BioClinicalBERT (a clinical adaptation of BioBERT from Alsentzer et al., 2019, through fine-tuning it on clinical text, which is in turn an adaptation of BERT itself through fine-tuning it on PubMed) and BiomedBERT (Gu et al., 2022, previously PubMedBERT, a model pre-trained from the ground up on PubMed) on the BERT side. Some honorable mentions include BioBART (Yuan et al., 2022), an adaptation of Facebook's BART; Med-PaLM 2 (Singhal et al., 2023), the medical adaptation of the second iteration of Google's PaLM (Chowdhery et al., 2022); and BLOOM (Workshop et al., 2023), a 176B-parameter fully open-access language model.

Along with the biomedical field, these models have also been adapted to a multilingual context in order to break language and cultural barriers, and offer greater flexibility and accessibility for users around the world. Some of these models include mBERT (multilingual BERT, Devlin et al., 2019) and mGPT (multilingual GPT, Shliazhko et al., 2023). One can imagine bridging the gap between the multilingual paradigm and the biomedical field in order to arrive at models that can perform well on multilingual biomedical related tasks, not just english ones.

In summary, research on Large Language Models like GPT and BERT has illuminated the amazing potential these models provide for interpreting natural language. The groundwork established by these LLMs becomes crucial when the fields of Named Entity Recognition and Entity Linking in the context of biomedical texts and in a multilingual paradigm are explored in the following sections.

2.4 Named Entity Recognition

As previously stated, Large Language Models are able to achieve state-of-the-art performance in many tasks. This is due to both the size of these types of models, and their architecture, which more often than not builds on the transformer architecture to deliver these results. The number of tasks that are well suited for LLMs is vast, and includes tasks like text generation (which can be further categorised depending on the style of text we want to generate), text summarization, text classification, language translation and question answering. However, the two tasks that this dissertation focuses on are Named Entity Recognition (NER) and Entity Linking (EL). This section will focus on the former while the following section will focus on the latter.

2.4.1 Definition and Motivation

Named Entity Recognition is an essential task in Natural Language Processing that involves identifying and categorising entity mentions in a given text, including names of individuals, groups, places, medical words, and other types of entities. For example, given the piece of unstructured text:

James went to Lidl and bought a Coca-Cola.

The aim is to produce the following annotations:

[James]$_{\text{Person}}$ went to [Lidl]$_{\text{Location}}$ and bought a [Coca-Cola]$_{\text{Product}}$.

This process is usually broken down conceptually into two distinct problems:

1. Detection: it is seen as a segmentation problem, where names are defined to be contiguous spans of tokens with no nesting (so for example, "Bank of America" is a single name despite "America" being itself a name).
2. Classification: the classification is done by the type of entity they refer to (e.g. person, organization, location, etc.), and it usually requires choosing a knowledge base by which to organize these categories.

Finding and classifying these entity mentions is NER's main goal, in order to obtain structured data from unstructured text. The categories are domain-specific, meaning that the researchers can develop their desired categories depending on the scope of the task. For example, in the context of biomedical documents, NER plays a pivotal role in extracting information about entities like genes, proteins, diseases, drugs, and other relevant biomedical terms and in integrating this biomedical data into knowledge bases. This in turn facilitates the extraction of valuable knowledge, aiding researchers and clinicians in a domain characterized by its specialized vocabulary, complex terminology, and diverse entities.

2.4.2 Evaluation

This task can be validated in a number of ways. For example, we can use accuracy and compare the predicted entity mentions against the gold standard, only judging a prediction as correct if the entity type and the span of the entity mention are correct. Another more robust way is to compute metrics like precision, recall and the F1-score. These metrics depend on the following concepts:

- True Positives (TP): number of correctly retrieved elements.
- False Positives (FP): number of incorrectly retrieved elements.
- False Negatives (FN): number of incorrectly not retrieved elements.

The metrics are then defined as follows:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

$$\text{F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

In other words, precision is the fraction of retrieved entity mentions that is relevant (correctly retrieved), where recall is the fraction of relevant mentions that was retrieved. This can be done at the entity type level (by separating them by type), or be measured at global scale. The F1-score is then the harmonic mean of the two. These measurements can be further applied in two ways: micro-average and macro-average. Micro-average is when the retrievals of all categories from the whole corpus are aggregated to compute the average metric, where as the macro-average is when the metric is computed independently for each category and then the average is taken. While micro-average gives equal weight to each instance and is useful when there is category imbalance, as it ensures that categories with more entity mentions do not have a greater impact on the overall metric, macro-average gives equal weight to each class regardless of its size, and is useful to evaluate the performance of the model on each separate category.

$$\text{Micro-Averaged Precision} = \frac{\sum_k \text{TP}_k}{\sum_k (\text{TP}_k + \text{FP}_k)}$$

$$\text{Micro-Averaged Recall} = \frac{\sum_k \text{TP}_k}{\sum_k (\text{TP}_k + \text{FN}_k)}$$

$$\text{Macro-Averaged Precision} = \frac{\sum_k \text{Precision}_k}{K}$$

$$\text{Macro-Averaged Recall} = \frac{\sum_k \text{Recall}_k}{K}$$

$$\text{Micro-Averaged F1-score} = \frac{2 \times \text{Micro-Averaged Precision} \times \text{Micro-Averaged Recall}}{\text{Micro-Averaged Precision} + \text{Micro-Averaged Recall}}$$

$$\text{Macro-Averaged F1-score} = \frac{2 \times \text{Macro-Averaged Precision} \times \text{Macro-Averaged Recall}}{\text{Macro-Averaged Precision} + \text{Macro-Averaged Recall}}$$

These are the standard metrics that shared tasks pick in order to evaluate their submissions, including the shared tasks addressed in the dissertation. Additionally, given the unique complexity of biomedical entities, other metrics can be introduced. For instance, the SNOMED CT Entity Linking Challenge[2] uses a character-level metric to account for the variations in token start and end indices.

2.4.3 Techniques

Historically, there are three major types of approach used to tackle this task. First, rule-based approaches rely on predefined linguistic patterns, regular expressions, and domain-specific rules to identify entities. For example, using regular expressions for phone numbers or a pattern like "X was born in Y" to categorize X as Person and Y as Date or Time. This approach can be applied to both the multilingual paradigm (Sekine & Nobata, 2004) and to specific domains (Chiticariu et al., 2010). However, the major drawback is that these types of approaches have very high maintenance costs and do not adapt well to new domains and languages (Mansouri et al., 2008).

Statistic-based approaches often include machine learning methods such as Conditional Random Fields (CRF, Settles, 2004), hidden Markov models (Morwal, 2012; Shen et al., 2003), or decision trees (Szarvas et al., 2006). These approaches leverage statistical models to capture the relationships within sequential data, but have disadvantages like their dependency on handcrafted features which affects performance if not done correctly (Eltyeb & Salim, 2014), or their inability to deal with data sparsity (Wuebker et al., 2013).

Additionally, statistic-based approaches are much less efficient at capturing contextual information than the current deep learning-based approaches. These approaches have grown in popularity over the last few years due to their increase in performance over the previous approaches. Two notable deep learning architectures employed in biomedical NER are Bidirectional Long Short-Term Memory networks with Conditional Random Fields (BiLSTM-CRF, Huang et al., 2015) and, more relevant to this dissertation, Large Language Models. Pre-trained LLMs have demonstrated state-of-the-art performance in biomedical NER by learning rich contextual representations of words and entities. Fine-tuning these models on biomedical datasets allows them to further adapt to the specific terminology and nuances of the biomedical domain. BERT and its offspring dominate this task, with various models like BiomedBERT or BioClinicalBERT pushing the state-of-the-art further.

2.4.4 Challenges

This task is not without its challenges, and those are further magnified by both the intricacies of the biomedical domain and the introduction of languages that are not English. As for the latter, a combination of diverse linguistic structures, language specific ambiguity and/or naming conventions, and lack of labeled data, all make building a model for less commonly spoken languages not a trivial task. On the other hand, the

[2] https://www.drivendata.org/competitions/258/competition-snomed-ct/page/817

ambiguity and complexity in common general domain free text alone poses a serious obstacle in and of itself, with the challenge of accurate identification of entity mentions in biomedical NER being even greater, as entities such as genes, proteins, diseases, and substances like drugs may have multiple names, abbreviations, acronyms, synonyms, and overall variations (Kundeti et al., 2016). Furthermore, this task does not fail only when an entity is not identified, or is misidentified. This is because an entity classification can also be partially correct. For instance, an entity can be correctly identified with fewer or more tokens than expected (wrong span), identifying more entity mentions than expected when adjacent (one vs. two entity mentions of type "Person" in Rodrigo Miguel), assigning an inexact type ("Substance" vs. "Drug") or identifying a mention based on a wrong scope, as pointed out by Marrero et al., 2013 (identifying "Alfredo da Costa" as a Person instead of "Maternidade Alfredo da Costa" as a Place). These examples demonstrate that the identification and classification processes in NER, specially when nuanced by a domain such as Biomedicine, are rather complex.

2.5 Entity Linking

2.5.1 Definition and Motivation

Entity Linking (EL) is a task that is closely related to Named Entity Recognition. While NER identifies and classifies entity mentions within the text based on their category, EL takes the whole process a step further by associating these recognized mentions with unique identifiers in some knowledge base, thus specifying exactly which entity it is. This task plays a pivotal role in enriching the extracted information from NER, as it bridges the gap between unstructured textual information and structured knowledge bases, and likely enables more profound insights into the entities mentioned. For instance, consider the already annotated example sentence from the last section:

$$[\text{James}]_{\text{Person}} \text{ went to } [\text{Lidl}]_{\text{Location}} \text{ and bought a } [\text{Coca-Cola}]_{\text{Product}}.$$

Given the above sentence as input, the Entity Linking system would output an unique identifier for both Coca-Cola and Lidl. Furthermore, it would also emit an unique identifier for James, disambiguating between the many James's that the text could be referring to. These identifiers are not only unique within that specific sentence - they are universal, meaning they can be used across different texts and datasets to consistently reference the same exact entity. This enables accurate linking on a global scale, no matter where or in what context the entity appears in.

2.5.2 Entity Linking In Combination With Named Entity Recognition

Named Entity Recognition and Entity Linking have a close relationship, as in order to link entities to a knowledge base, the mentions to those entities have to be identified first. Thus, they are regularly treated as two steps in a bigger information extraction and disambiguation pipeline: first, NER extracts and categorizes entity mentions from unstructured text based on categories, providing a foundational understanding of the content; subsequently, EL enriches this information by associating mentions with precise identifiers. This linkage not only disambiguates mentions but also facilitates deeper insights into the entities mentioned, enabling a more nuanced interpretation. According to French and McInnes (2023), biomedical EL systems in particular were exclusively jointly performed with NER, which changed "in the mid-2010s with the release of prominent entity linking corpora (...). (...) For the first time, researchers could model BEL as an independent task". However, it is still common for NER to act as the precursor to EL. For example, Martins et al. (2019) illustrate the benefit of joint learning of NER and EL with the following example:

Leeds’ Bowyer fined for part in fast-food fracas.		
	NER	EL
Separate	Leeds-ORG	Leeds
Joint	Leeds-ORG	Leeds_United_A.F.C.

Figure 2.7: Example of the potential benefits of joint learning. Wrong entity in red and correct in green. Adapted from Martins et al., 2019.

This pairing of NER and EL is crucial for unlocking the full potential of information extraction systems, particularly in the complex and nuanced domain of biomedical documents across multiple languages.

2.5.3 Knowledge bases

While in open-domain Entity Linking it is common to use knowledge bases derived from Wikipedia, biomedical EL relies on key domain-specific knowledge bases to link the entity mentions to. The best example of such is the Unified Medical Language System[3] (UMLS, Bodenreider, 2004), which plays a central role by integrating diverse biomedical vocabularies and providing a unified framework for consistent entity representation. Within UMLS, knowledge bases like Medical Subject Headings[4] (MeSH), Online Mendelian Inheritance in Man[5] (OMIM) and DrugBank[6] significantly contribute to the richness and breadth of biomedical knowledge represented in UMLS. MeSH, developed by the US National Library of Medicine, aids in consistent indexing of biomedical literature (predominantly from PubMed) and is foundational for general biomedical concepts. OMIM specializes in genetic disorders and traits, thus being invaluable for understanding the genetic basis of diseases and traits. Likewise, DrugBank also has a narrower focus than MeSH as it serves

[3] https://www.nlm.nih.gov/research/umls/index.html

[4] https://www.ncbi.nlm.nih.gov/mesh/

[5] https://www.ncbi.nlm.nih.gov/omim/

[6] https://go.drugbank.com

as a dedicated resource for pharmaceutical information, enriching Entity Linking with details on drug interactions, pharmacology, and molecular targets. These are examples of knowledge bases that provide a comprehensive foundation for mapping entities in biomedical texts. The knowledge bases behind the shared tasks that will be used for evaluation will be covered in their respective sections.

2.5.4 Evaluation

Evaluating the performance of an EL system is somewhat similar to a NER system: while the micro and macro versions of precision, recall and F1-score can be used, accuracy is usually the chosen metric. In Entity Linking, accuracy measures how often the system correctly links an entity mention to the correct entity (usually a code) in the knowledge base. For this purpose, the Accuracy@1 (or just accuracy) is often chosen, which evaluates the proportion of times that the first suggestion made by the system (the top-ranked candidate), and only the first suggestion, is the correct entity.

However, it is not uncommon for the top suggestion to be incorrect, and at the same time relevant candidates to still be included in the systems' top suggestions. For scenarios like these, the Accuracy@N becomes useful, as it measures the proportion of times the correct entity appears within the top N candidates suggested by the system, not just the first one. This is particularly valuable in practical applications where downstream users or systems can consider multiple suggestions, increasing the likelihood of identifying the correct entity even if the top suggestion is incorrect.

The formulas for these metrics are as follows:

$$\text{Accuracy@1} = \frac{\text{Number of correct predictions in top-1}}{\text{Total number of predictions}}$$

$$\text{Accuracy@N} = \frac{\text{Number of correct predictions in top-N}}{\text{Total number of predictions}}$$

2.5.5 Techniques

Similarly to evaluation, the approaches taken to perform this task are essentially the same when compared to NER. First, rule-based approaches which involve predefined sets of rules that guide the linking process, rules such that may consider entity characteristics, context, or similarity like the system developed by Leal et al. (2015), which minimizes the Levenshtein distance to SNOMED-CT candidates.

Statistic-based approaches leverage statistical models and probabilities to infer the most likely link between an identified entity and a knowledge base entry, one of the best known early attempts being DNorm, by Leaman et al. (2013), which employed a statistic called Term Frequency-Inverse Document Frequency and was later incorporated into TaggerOne for joint NER and EL (Leaman & Lu, 2016).

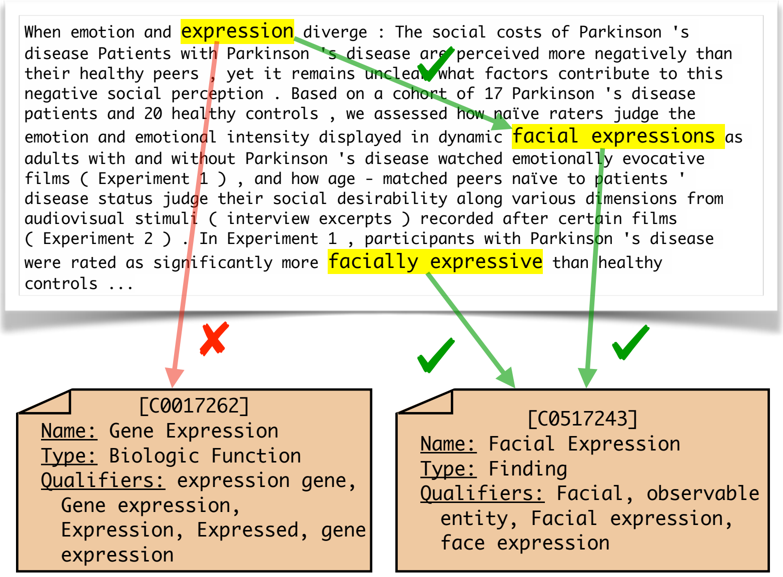
Deep learning-based approaches have unsurprisingly shown remarkable success in Entity Linking by now, both in open-domain and domain-specific applications, namely Biomedicine. Pre-trained Large Language Models such as BERT and its variants like BiomedBERT are able to capture contextual representations from large biomedical corpora and thus enable state-of-the-art disambiguation and linking of entity mentions. SapBERT, or Self-aligning pre-trained BERT, (Liu, Shareghi, et al., 2021), another adaptation of the original BERT model, achieved a new state-of-the-art on multiple benchmarks by proposing a pre-training scheme that self-aligns the representation space of biomedical entities, therefore tackling the challenge of accurately capturing fine-grained semantic relationships in the biomedical domain. This approach was later adapted to a multilingual scenario (Liu, Vulić, et al., 2021), yielding consistent gains across a multitude of languages. Guven and Lamurias (2023) explore surface-based, approximate nearest neighbour search and embedding approaches during candidate generation along with pre-trained LLMs, including BioBERT and SapBERT, to develop a framework for multilingual biomedical Entity Linking, and highlight the challenges that come with domain-specific multilingual datasets by underperforming when compared to state-of-the-art monolingual approaches. BLINK[7], developed by researchers at Facebook (Wu et al., 2020), was incorporated by Guven and Lamurias (2023) in their research as it is a popular approach that achieves state-of-the-art by using a two stage approach for Entity Linking, based on fine-tuned BERT architectures.

Lastly, another type of approach that EL introduces as opposed to NER, given its nature, is graph-based approaches. These methods model entities (nodes) and their relationships (edges) as graphs, and then use graph algorithms to analyze them and determine the optimal links for the entity mentions based on the semantic connections. Angell et al. (2021) propose a model in which linking decisions can be made by grouping multiple mentions together via clustering and jointly making linking predictions, thus accounting for the relationships within and across documents between entities, especially ones that have either a generic or a highly specialized form. Another example is Ruas et al. (2020) approach, which builds a disambiguation graph where the nodes are the normalization candidates (from the knowledge base) for the entity mentions and the edges are the entity relationships established in the text. Then, the Personalized PageRank algorithm is used to choose the candidates that maximises the coherence of the disambiguation graph.

2.5.6 Challenges

Entity ambiguity is one of the major challenges in Entity Linking, further accentuated when we step into the biomedical domain. For example, the mention *expression* of the entity *Facial Expression*, in the context of Figure 2.8, is highly ambiguous compared to the other mentions, and could easily be confused with the more prevalent entity *Gene expression* (Angell et al., 2021).

[7] https://github.com/facebookresearch/BLINK



When emotion and expression diverge : The social costs of Parkinson 's disease Patients with Parkinson 's disease are perceived more negatively than their healthy peers , yet it remains unclear what factors contribute to this negative social perception . Based on a cohort of 17 Parkinson 's disease patients and 20 healthy controls , we assessed how naïve raters judge the emotion and emotional intensity displayed in dynamic facial expressions as adults with and without Parkinson 's disease watched emotionally evocative films (Experiment 1) , and how age - matched peers naïve to patients ' disease status judge their social desirability along various dimensions from audiovisual stimuli (interview excerpts) recorded after certain films (Experiment 2) . In Experiment 1 , participants with Parkinson 's disease were rated as significantly more facially expressive than healthy controls ...

[C0017262]
Name: Gene Expression
Type: Biologic Function
Qualifiers: expression gene, Gene expression, Expression, Expressed, gene expression

[C0517243]
Name: Facial Expression
Type: Finding
Qualifiers: Facial, observable entity, Facial expression, face expression

Figure 2.8: An example of entity ambiguity. All three highlighted mentions refer to the same entity *Facial Expression*, but the mention *expression*, if considered independently, could be mistaken for the incorrect entity *Gene Expression*. Adapted from Angell et al., 2021.

In addition to ambiguity, biomedical entities often exhibit variations in naming conventions (Sung et al., 2020). Entities such as genes, proteins, and diseases, may have multiple names, acronyms, abbreviations, and synonyms. For instance, as described by Liu, Shareghi, et al. (2021), "the medication Hydroxychloroquine is often referred to as Oxichlorochine (alternative name), HCQ (in social media) and Plaquenil (brand name)". Resolving these variations by considering the broader textual context requires sophisticated disambiguation techniques.

Another challenge in biomedical EL is the fact that biomedical research continually introduces new entities, especially in emerging fields. This poses two problems: first, knowledge bases may not always encompass the entirety of these entities, in which case the model has to cope with their absence in some way; and second, the model may not be able to deal with entities that were introduced into the knowledge base after training, in which case regular retraining is required to keep the model updated. This in turn gives rise to another problem, which is that EL systems are often very dependent of the knowledge base used and do not translate well to other knowledge bases.

Lastly, Entity Linking in the multilingual domain also presents some challenges (French & McInnes, 2023): ambiguities in translations, variations in naming conventions, and differences in linguistic structures are obstacles to take into account. The task becomes even more intricate due to the possibility of a single term having multiple translations and refer to different entities, or terms that exist in some languages but not in others. Furthermore, the scarcity of annotated data for less commonly spoken languages can hinder the development of robust multilingual EL systems.

In summary, the sections on Named Entity Recognition and Entity Linking provide a comprehensive understanding of the methodologies and challenges in extracting and associating entity mentions in biomedical texts. The subsequent exploration of Transfer Learning and Data Augmentation will further advance the precision and adaptability of these processes.

2.6 Transfer Learning and Data Augmentation

In the ever-evolving landscape of Natural Language Processing, the integration of advanced techniques such as Transfer Learning and Data Augmentation has emerged as a strategy to enhance the precision and adaptability of information extraction processes.

2.6.1 Transfer Learning

Transfer Learning, as it was explained in the section titled Pre-Training and Fine-Tuning, enables models to leverage pre-existing knowledge. It involves reusing the knowledge gained from one task to enhance performance on a different but related task. This technique significantly improves learning efficiency, as the model does not have to be trained from scratch for every single application. A simple open-domain example is when training a classifier to predict whether or not an image contains food, you could use the knowledge it gained during training to also recognize drinks. This in turn gave rise to the so-called Foundation Models, pre-trained on massive amounts of generic unlabeled data that can be fine-tuned to just about any task with the available annotated data.

Additionally, biomedical knowledge expressed in high-resource language knowledge bases has been successfully transferred into LLMs tuned for low-resource languages. For instance, Souza et al. (2020b) approach of fine-tuning BERT to the Portuguese Language, together with a CRF for the final label prediction was the first use of the BERT model in the Portuguese NER task and improved the state-of-the art for the HAREM I dataset. Likewise, Boudjellal et al. (2021) approach that investigates the effectiveness of pre-training a monolingual BERT model with a small-scale Arabic biomedical dataset outperformed both AraBERT and multilingual BERT, two state-of-the-art methods.

Even more impressively, Liu, Vulić, et al. (2021) proposed an improvement over SapBERT (Liu, Shareghi, et al., 2021), a state-of-the-art fine-tuning method that aligns BERT's representations with its UMLS synonyms to improve BERT's performance. The improvement comprises of 1) leveraging multilingual UMLS synonyms instead of only English ones, and 2) introducing general-domain translation knowledge (i.e. pairing general-domain sentences, translated from one language to another, and feeding it to the model) to help with resource-poor languages where domain-specific knowledge is scarce, but general-domain translation knowledge is more readily available. This last strategy helped propagate the available English knowledge to those resource-poor languages. Their proposed methods yielded consistent gains across all target languages. These examples illustrate not only the motivation behind Transfer Learning (only needing a relatively small set of domain-specific data) but also its effectiveness.

2.6.2 Data Augmentation

Data Augmentation addresses challenges related to limited labeled data. In the multilingual biomedical domain, the scarcity of labeled data remains a significant bottleneck (Almeida et al., 2023), with manual annotation often being impractical and resource intensive, and therefore not a suitable strategy for curating a sufficiently large dataset. Thus, Data Augmentation allows us to further expand datasets with new data by applying transformations to the existing data. Augmenting the training data introduces diversity, exposing the model to different ways entities can manifest, and improving its generalization ability and adaptability to unseen instances.

There are a number of data augmentation techniques, not only for text data but also for images, such as geometric and color transformations, and for audio, like noise injection or changing speed and pitch. However, textual data augmentation is composed by techniques like word shuffling (Su et al., 2021) and word replacement, usually by a synonym (Su et al., 2021), but variations exist: Almeida et al. (2023) replaced random tokens with the `[UNK]` (unknown) token to force the model to consider the surrounding context, while Phan and Nguyen (2022) replaced a token with a semantically similar one and then evaluated the resulting sentence to verify if it is still in fact semantically similar. Other techniques include word insertion/deletion (Su et al., 2021), paraphrasing (e.g., with a generative model like Guimarães et al., 2024), translation and back translation (Pappas et al., 2022).

By leveraging pre-existing knowledge and augmenting training datasets, biomedical NLP across languages can be enhanced by innovative strategies that allow it to overcome challenges such as limited labeled data and domain-specific variations. Both transfer learning and data augmentation techniques are thus explored in this dissertation.

2.7 Shared Tasks

Shared tasks play a pivotal role in advancing biomedical Information Extraction, by providing standardized benchmarks and fostering collaboration. The motivation behind participating in shared tasks goes beyond competition: they serve as platforms where researchers converge to evaluate and compare their approaches in a controlled environment, pushing the state-of-the-art forward.

Biomedical Text Mining, in particular the Spanish branch, offers a substantial amount of shared tasks, each one with different objectives, or focusing on different types of entities. Tasks such as ProfNER (Miranda-Escalada et al., 2021), whose goal is the identification of professions and occupations in Spanish health related tweets; MEDDOCAN (Marimon et al., 2019), specifically devoted to the anonymization of medical documents in Spanish; or MedProcNER/ProcTEMIST (Lima-López, Farré-Maduell, Gascó, et al., 2023), dedicated to Named Entity Recognition on clinical procedures, illustrate some of the themes that these tasks can encompass. Furthermore, some more recent tasks also offer entity extraction and linking for a multitude of languages. This dissertation focuses on three shared tasks:

SympTEMIST, CANTEMIST and MultiCardioNER.

2.7.1 SympTEMIST

SympTEMIST (Lima-López, Farré-Maduell, Gasco-Sánchez, et al., 2023), short for "SYMPtoms, signs and findings TExt MIning Shared Task", focuses on the detection and normalization of symptoms, signs and findings in medical documents in Spanish, as the name suggests. After detection, the entity mentions are normalized to SNOMED CT[8], a multilingual clinical healthcare terminology that "enables consistent representation of clinical content in electronic health records". It is made up three subtasks:

- SymptomNER - automatic detection of mention spans of symptoms (including signs and findings) from clinical reports written in Spanish from the SympTEMIST corpus, a collection of 1,000 clinical case reports in Spanish.
- SymptomNorm - automatic normalization (linking) of symptom mentions in Spanish to their corresponding SNOMED CT concept identifiers.
- SymptomMultiNorm - automatic normalization (linking) of symptom mentions in a variety of languages - namely English, French, Portuguese, Dutch and Italian - to their corresponding SNOMED CT concept identifiers. This subtask is experimental, and those corpora were derived from the automatic translation of most of the documents from the Spanish corpus, but not all of them. To make this distinction clear, the datasets that correspond to this subtask will not be identified with the suffix EL, but with the suffix MEL (SympTEMIST-MEL), meaning Multilingual Entity Linking.

The primary evaluation metrics consist of the micro-averaged precision, recall and F1-scores for the NER subtask, and the accuracy for the two EL subtasks. Tables 2.1, 2.2 and 2.3 present the most significant results previously obtained for each of the subtasks, with the mean and median values also being reported at the bottom of each table.

Table 2.1: Best and second best results for each metric, along with their mean and median, from the SymptomNER subtask.

Team Name	Precision	Recall	F1-score
ICB	**0.8039**	0.6988	**0.7477**
ICB	0.7895	0.7068	0.7459
FRE	0.7154	**0.7403**	0.7277
FRE	0.7231	0.7303	0.7267
Mean	0.6639	0.6233	0.6420
Median	0.7181	0.6885	0.7054

[8] https://www.snomed.org

Table 2.2: Best and second best accuracies, along with the mean and median, from the SymptomNorm subtask.

Team Name	Accuracy
HPI-DHC	**0.6070**
BIT.UA	0.5890
Fusion@SU	0.5890
Mean	0.4928
Median	0.5321

Table 2.3: Best and second best accuracies for each language, along with the mean and median, from the SymptomMultiNorm subtask.

Language	Team Name	Accuracy
EN	BIT.UA	**0.7250**
	BIT.UA	0.7137
	Mean	0.6725
	Median	0.7193
FR	BIT.UA	**0.5733**
	BIT.UA	0.5726
	Mean	0.5282
	Median	0.5726
IT	BIT.UA	**0.6703**
	BIT.UA	0.6697
	Mean	0.5416
	Median	0.5421
NL	BIT.UA	**0.6397**
	BIT.UA	0.6389
	Mean	0.5967
	Median	0.6353
PT	BIT.UA	**0.5575**
	BIT.UA	0.5569
	Mean	0.4995
	Median	0.5555

2.7.2 CANTEMIST

CANTEMIST (Miranda-Escalada et al., 2020), short for "CANcer TExt Mining Shared Task", is dedicated to the extraction and normalization of tumor morphology. It relies on the CANTEMIST corpus, composed of 3000 manually selected clinical cases of different cancer types. The entity mentions are normalized on the International Classification of Diseases for Oncology[9] (specifically the third revision, ICD-O-3), a medical classification maintained by the World Health Organization. This task is structured into three independent subtasks:

- CANTEMIST-NER - automatic detection of tumor morphology mentions from the CANTEMIST corpus.
- CANTEMIST-NORM - automatic normalization of tumor morphology, that requires all tumor morphology entity mentions together with their corresponding ICD-O-3 codes.
- CANTEMIST-CODING - requires returning a ranked list of the corresponding ICD-O-3 codes for each document.

The official evaluation metric is the micro-averaged F1-score for all subtasks. However, the CANTEMIST-NORM subtask was not covered in this dissertation due to lack of time, and the CANTEMIST-CODING subtask is not relevant for the general theme of NER and EL. Thus, Table 2.4 presents the most significant results from the CANTEMIST-NER task.

Table 2.4: Best and second best results for each metric, along with their mean and median, from the CANTEMIST-NER subtask.

Team Name	Precision	Recall	F1-score
HITSZ-ICRC	**0.8710**	0.8680	**0.8700**
Vicomtech	0.8680	**0.8710**	0.8690
Mean	0.7300	0.7400	0.7280
Median	0.8160	0.7860	0.8010

2.7.3 MultiCardioNER

MultiCardioNER (Lima-López et al., 2024) focuses on the adaptation of clinical NER systems to the cardiology domain. To achieve this, each task targets one of two key clinical concept types - diseases and medications - each annotated over the same set of documents. Then, the extraction of those types of entities is specifically performed over cardiology clinical case documents. It is composed of the following two subtasks:

[9] https://www.who.int/standards/classifications/other-classifications/international-classification-of-diseases-for-oncology

- CardioDis - disease recognition in Spanish text. For this, the DisTEMIST corpus, which was already used on a previous subtask (Miranda-Escalada et al., 2022), is provided as training data. An additional development set of cardiology clinical case reports is also provided for further fine-tuning.

- MultiDrug - medication recognition in three languages: Spanish, English and Italian. The newly-released DrugTEMIST corpus is provided in those three languages to train the systems, and again an additional development set of cardiology clinical case reports is available for the fine-tuning process.

The main evaluation metrics are the micro-averaged precision, recall and F1-scores. Tables 2.5 and 2.6 display the most relevant official results for the MultiDrug and CardioDis subtasks, respectively, including some of our own submissions (NOVALINCS) which achieved great precision scores. However, those are not the final scores that we present in the Results and Discussion section regarding this task, as an error in obtaining the predictions hindered the recall in almost every one of our submissions, which resulted in poor official F1-scores comparatively to the other submissions. Nevertheless, to keep the theme of best and second best reported results displayed for each metric, we included our original submission's scores in the aforementioned tables.

Table 2.5: Best and second best results for each metric for each language, along with their mean and median, from the MultiDrug subtask.

Language	Team Name	Precision	Recall	F1-score
ES	ICUE	0.9146	**0.9412**	**0.9277**
	Enigma	0.9130	0.9348	0.9238
	Enigma	0.9148	0.9005	0.9076
	NOVALINCS	**0.9242**	0.4965	0.6460
	Mean	0.8143	0.7057	0.7316
	Median	0.8853	0.8824	0.8502
EN	Enigma	0.8981	**0.9477**	**0.9223**
	ICUE	**0.9086**	0.9128	0.9107
	Enigma	0.9031	0.8989	0.9010
	ICUE	0.8314	0.9343	0.8799
	Mean	0.8373	0.7457	0.7723
	Median	0.8692	0.8983	0.8769
IT	Enigma	0.8840	0.8844	**0.8842**
	Enigma	0.8723	0.8956	0.8838
	Enigma	0.9016	0.8606	0.8806
	ICUE	**0.9114**	0.8461	0.8776
	ICUE	0.8186	**0.9000**	0.8574
	Mean	0.7842	0.7059	0.7195
	Median	0.8162	0.8519	0.8421

Table 2.6: Best and second best results for each metric, along with their mean and median, from the CardioDis subtask.

Team Name	Precision	Recall	F1-score
BIT.UA	0.8155	**0.8243**	**0.8199**
BIT.UA	0.8110	0.8181	0.8145
PICUSLab	**0.8886**	0.4744	0.6185
NOVALINCS	0.8183	0.3398	0.4802
Mean	0.7397	0.6862	0.6956
Median	0.7811	0.7459	0.7229

3 Methodology and Experiments

This chapter describes the methodology and experimental setup used in this dissertation to address the challenges of Named Entity Recognition (NER) and Entity Linking (EL) in the multilingual biomedical domain. We begin by briefly highlighting the third-party tools that helped us put our research into practice. Following that, we give a thorough description and statistical analysis of the data provided by the shared tasks we chose to evaluate our approaches in. Finally, we dive into the main focus of this chapter and discuss the specifics of both our NER and EL strategies, dividing them into two sections each, the first regarding Transfer Learning and the second Data Augmentation.

3.1 Third-Party Tools

In todays Natural Language Processing (NLP) landscape, state-of-the-art models are growing at an exorbitant pace. Working with such large and complex models requires robust tools not just to facilitate, but enable experimentation to begin with. To train and fine-tune these models in an effective way, it is crucial to use a combination of such tools and services, taking advantage of both 1) massive computational resources and 2) specialized frameworks and libraries, that facilitate the use of such models and speed up the development process. Thus, we briefly highlight below the third-party tools and resources that enabled us to perform the research presented later in this section.

Throughout the Related Work section, various models were mentioned, with the likes of BiomedBERT (Gu et al., 2022) for the biomedical domain, or XLM-RoBERTa (Conneau et al., 2020) for multilingual applications, being popular choices. To take advantage of such models, we used the Hugging Face platform. Hugging Face[1] is a company that specializes in several areas of machine learning like computer vision, audio processing, and critically, Natural Language Processing. It provides a popular open-source platform for working with and sharing state-of-the-art models, with their open-source Transformers library providing an extensive collection of pre-trained transformer models (as well as tools for working with them) and supporting a wide range of tasks including Named

[1] https://huggingface.co

Entity Recognition and Entity Linking. The platform has become a central hub not only for Natural Language Processing, but for Artificial Intelligence as a whole, and it underpins the work done in this dissertation in the sense that, not only were all of the pre-trained models that we used available to us because of their platform, but we also published our final fine-tuned models, as well as the parsed datasets, for public and ease of use. Additionally, our NER fine-tuning process was heavily inspired by their scripts and relied on their own python libraries.

It is also worth mentioning the two sources of computational resources that allowed us to perform our research with such large models: Google Colab and the DI-Cluster. Google Colab[2] is a cloud-based platform that allows users to write, run, and share Python code. It leans heavily into machine learning, data analysis, and deep learning related activities, providing free access to powerful computing resources like the current best NVIDIA GPUs. The DI-Cluster is a computing cluster managed by the Department of Computer Science of NOVA School of Science and Technology. It provides a free, powerful and easy to use computing platform for students and researchers. Both of these platforms were used extensively throughout our research.

3.2 Data

In this section, we present the data used in our experiments, which forms the foundation for fine-tuning and evaluating the models. The data was sourced from the three previously mentioned Shared Tasks, providing conceptually rich and idiomatically diverse data, necessary for developing robust solutions for the multilingual biomedical entity identification and linking problems. We provide an overview of the data, detailing its origins and characteristics, and present key statistics that highlight the scope and scale of the datasets. This contextual background is useful for understanding our experiments and results.

3.2.1 SympTEMIST

The corpus provided by the SympTEMIST shared task (Lima-López, Farré-Maduell, Gasco-Sánchez, et al., 2023) is a collection of 1,000 documents that belong to the Spanish Clinical Case Corpus (SPACCC Intxaurrondo & Krallinger, 2018), a manually classified collection of clinical case reports all written in the Spanish language, and spanning various different medical specialties. These documents are then divided into 750 documents for training, and 250 for testing. On the Spanish EL track, only a subset of 450 of the documents of the train set are used, and only 350 on the various MEL tracks. As the name of this shared task suggests, it focuses on symptom-type entities, with SINTOMA being the designated entity type. Table 3.1 presents some statistics regarding the number of documents, and

[2] https://colab.google

unique and total annotated entity mentions in both the training and test sets, from each subtask.

Table 3.1: The number of documents, unique mentions, and total mentions, from each SympTEMIST subtask.

Dataset	Language	Split	Docs	Unique Mentions	Total Mentions
SympTEMIST-NER	ES	Train	750	6,177	9,092
	ES	Test	250	2,388	3,104
SympTEMIST-EL	ES	Train	450	2,507	3,484
	ES	Test	250	2,144	2,848
SympTEMIST-MEL	EN	Train	350	1,126	2,003
	EN	Test	250	903	1,600
	FR	Train	350	1,024	1,802
	FR	Test	250	856	1,425
	IT	Train	350	1,080	1,924
	IT	Test	250	905	1,544
	PT	Train	350	896	1,686
	PT	Test	250	909	1,521

Regarding the Entity Linking tracks (subtasks 2 and 3), the number of unique and total SNOMED CT codes is shown in Table 3.2, providing insights into the dataset's balance.

Table 3.2: The number of unique and total SNOMED CT codes, from each SympTEMIST Entity Linking subtask.

Dataset	Language	Split	Unique Codes	Total Codes
SympTEMIST-EL	ES	Train	1,535	3,484
	ES	Test	1,391	2,848
SympTEMIST-MEL	EN	Train	841	2,003
	EN	Test	704	1,600
	FR	Train	780	1,802
	FR	Test	668	1,425
	IT	Train	823	1,924
	IT	Test	708	1,544
	PT	Train	671	1,686
	PT	Test	690	1,521

Additionally, this shared task also includes an extensive gazetteer, created with the assistance of SNOMED CT, mapping entity mentions to their respective SNOMED CT codes. It includes a set of 164,817 lexical entries, comprising 121,713 unique codes.

3.2.2 CANTEMIST

The CANTEMIST (Miranda-Escalada et al., 2020) corpus is a collection of 1,301 oncological clinical case reports written in Spanish. It is split into four subsets: the training subset containing 501 documents, two development subsets totalling 500 documents, and the test subset with 300 documents. All documents have been manually annotated by clinical experts with mentions of tumor morphology, designated as MORFOLOGIA_NEOPLASIA. The numbers of documents, and unique and total entity mentions for each split are presented in Table 3.3.

Table 3.3: The number of documents, unique mentions, and total mentions, from each CANTEMIST-NER split.

Dataset	Split	Docs	Unique Mentions	Total Mentions
CANTEMIST-NER	Train	501	2,096	6,396
	Dev	500	2,286	6,001
	Test	300	1,295	3,633

As explained in the Shared Tasks section, despite CANTEMIST's EL subtask being in the scope of this dissertation thematically, we ultimately did not work on it due to time constraints. Moreover, we felt that we already had a sufficient body of existing EL results which could provide a comprehensive benchmark for future research. Thus, we only used CANTEMIST's NER annotations.

3.2.3 MultiCardioNER

The MultiCardioNER shared task (Lima-López et al., 2024) utilizes two primary datasets: DisTEMIST (Miranda-Escalada et al., 2022) and DrugTEMIST. Both of these datasets encompass the same 1,000 documents from the SPACCC corpus, just like the previously mentioned SympTEMIST corpus. This time, though, they are all used as the training corpus and are not divided into training and testing. While DisTEMIST is focused on extracting disease mentions, thus having ENFERMEDAD as its sole and primary label, DrugTEMIST focuses on drug and medication extraction, with FARMACO being its entity type. Table 3.4 displays the number of documents, and unique and total entity mentions for each dataset.

Furthermore, since this task is concerned with the adaptation of clinical NER systems to the cardiology domain, a separate cardiology clinical case reports dataset is also provided, named CardioCCC. It contains a total of 508 documents, split into 258 documents initially intended for validation, and 250 for testing. As expected, these documents only include annotations for disease and medication mentions, the entity types considered for this competition. Table 3.5 presents statistics on the documents, and unique and total entity mentions for each split of this dataset.

Table 3.4: The number of documents, unique mentions, and total mentions, from each MultiCardioNER dataset.

Dataset	Language	Docs	Unique Mentions	Total Mentions
DisTEMIST	ES	1,000	6,739	10,664
DrugTEMIST	ES	1,000	925	2,778
	EN	1,000	875	2,814
	IT	1,000	893	2,808

Table 3.5: The number of documents, unique mentions, and total mentions, from each CardioCCC subset.

Dataset	Entity Type	Language	Docs	Unique Mentions	Total Mentions
CardioCCC - Dev	Diseases	ES	258	4,749	10,348
	Drugs	ES	258	526	2,510
	Drugs	EN	258	513	2,510
	Drugs	IT	258	520	2,585
CardioCCC - Test	Diseases	ES	250	3,784	7,884
	Drugs	ES	250	465	1,718
	Drugs	EN	250	460	1,721
	Drugs	IT	250	469	1,800

In the next section, we will explore the Named Entity Recognition experiments that we conducted, analyzing the aspects of both Transfer Learning and Data Augmentation. Following this, we will do the same for Entity Linking.

3.3 Named Entity Recognition Experiments

The current section presents the Named Entity Recognition experiments we ran as part of our research. Following the previous discussion about the data we gathered from the shared tasks, we now concentrate on the procedures for achieving the results we will later present. This section is divided into two main parts: Transfer Learning and Data Augmentation. In the transfer learning portion, we start by 1) giving context on the fine-tuning process applying an encoder-based model to this kind of task; next, we 2) explore the selection of the models used, 3) give some context on our participation in the MultiCardioNER shared task, 4) cover the necessary preprocessing steps we took for parsing the datasets, and finally 5) the structured fine-tuning pipeline that was implemented. Then, the data augmentation part will demonstrate how we leveraged word2vec models to try to enhance the fine-tuning process.

3.3.1 Transfer Learning

3.3.1.1 Fine-tuning for NER

Fine-tuning for the task of Named Entity Recognition involves adapting a pre-trained language model to predict entity labels in a given labeled dataset. This dataset consists of sentences where each token (typically a word) is tagged with a label indicating whether it is part of an entity mention or not. A tagging format that is commonly used for this is the IOB2 format. IOB, short for "Inside, Outside, Beginning", was introduced by Ramshaw and Marcus (1995), and when applied to NER, it takes the following meaning:

- The "B-" prefix on a tag indicates that the token corresponds to the beginning of an entity mention.
- The "I-" prefix indicates that the token belongs inside of an entity mention.
- An "O" tag indicates that a token does not belong to an entity mention.

Listing 3.1 shows an example of this tagging scheme.

```
He              O
was             O
admitted        O
with            O
treatment       O
with            O
intravenous     O
amoxicillin     B-FARMACO
-               I-FARMACO
clavulanic      I-FARMACO
acid            I-FARMACO
.               O
```

Listing 3.1: An example of the IOB tagging scheme used to identify the drug (FARMACO in Spanish) "amoxicillin-clavulanic acid".

This is what the model is tasked with learning during the fine-tuning process, to tag entity mentions of a given type present in free-text, while leveraging all of its pre-existing pattern knowledge of the language(s) it was pre-trained on.

The way the model learns how to correctly predict these mentions is through trying to minimize the loss function. Typically, the loss function used for token classification problems, more specifically NER, is the Cross-Entropy Loss:

$$\mathcal{L} = -\frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{C} y_{i,j} \log(p_{i,j})$$

This loss function works in the following way:

- At its final layer, the model generates a set of predictions for each token in a given sequence (the input sentence). For NER, these predictions take the form of probability distributions over the possible entity labels. For example, in the IOB2 tagging scheme, where C is the number of possible entity labels (in this case, 3: "B-", "I-", "O"), the layer outputs a 3-dimensional vector for each token. Each value in these vectors represents the probability that the token belongs to the beginning of an entity mention, inside of a mention, or no mention at all.

- Then, the Cross-Entropy Loss function $\mathcal{L}$ is used to compare the predicted probability distributions $p_{i,j}$ with the actual labels $y_{i,j}$ for each token i. The correct labels are encoded as one-hot vectors, where only one element, corresponding to the correct class, is set to 1, and the others are 0. The loss penalizes the model when it assigns high probabilities to incorrect labels or low probabilities to the correct label. The logarithm of the predicted probability, $\log(p_{i,j})$, is used to penalize the model even more severely when it predicts a very low probability for the correct label.

- The overall loss $\mathcal{L}$ is computed as an average over all N tokens in the sequence. After doing this for a batch of sequences, the model's parameters are updated through backpropagation and the optimizer to try and minimize this average loss, i.e. assign higher probabilities to the correct labels.

To summarize, the loss function, particularly Cross-Entropy Loss, plays a central role in fine-tuning a model for Named Entity Recognition, by quantifying prediction errors and providing the necessary gradients for backpropagation. Through this process, the model becomes increasingly adept at recognizing and classifying entity mentions. This background is important for better understanding the subsequent steps.

3.3.1.2 Pre-trained LLMs Chosen

We started by searching for a Spanish state-of-the-art biomedical model to tackle the Spanish part of the MultiCardioNER shared task, as this task was taking place at the same time that we started focusing on Named Entity Recognition. We found bsc-bio-ehr-es (Carrino et al., 2022), a BERT-based model with a focus on the biomedical domain. It has been trained on more than 1B tokens of Spanish content related to the biomedical domain, including a crucial collection of more than 278K clinical documents, which aligns perfectly with this shared task's motto of the exploration of clinical NER systems. Because the authors of the model belong to the same organization that set up this shared task, we confirmed with the task organization that the CardioCCC corpus was not used in the pre-training of this model, so the possibility of test data leakage is ruled out. We also decided to use this model for the remaining Spanish shared tasks, SympTEMIST and CANTEMIST. However, upon further investigation, we realized that it had already been fine-tuned on the CANTEMIST shared task by its creators, in order to evaluate its performance. Nevertheless, we still applied this model to said task in the same way that we applied it to the others, if anything to have another means of comparing our approach.

Having chosen the model to tackle the Spanish subtasks, we focused on the other languages that the MultiCardioNER task covers: English and Italian. For the English subtask, we went with BioLinkBERT (Yasunaga et al., 2022), the current best model for NER according to BLURB (Gu et al., 2022), a benchmark for English biomedical models. Unfortunately, due to resource limitations, we were only able to use the base version, which nevertheless still yields very similar results according to the BLURB. For the Italian subtask, we chose bioBIT (Buonocore et al., 2023), a model built on top of Italian BERT, further trained on data derived from machine-translated biomedical abstracts. Between bioBIT and medBIT (a similar model by the same authors, except that it is further pre-trained on a small corpus of medical textbooks), the difference in performance on NER tasks is negligible: while bioBIT excelled when evaluated on certain NER subtasks, medBIT performed better in others, demonstrating that both models have their strengths depending on the specific nuances of the task. Given this, we decided to proceed with bioBIT as it is the most widely used. Table 3.6 presents each of the models and the dataset they were applied to.

Table 3.6: The models (their Hugging Face repository IDs) chosen for the Named Entity Recognition experiments, per language.

Dataset	Language	Model
SympTEMIST-NER	ES	PlanTL-GOB-ES/bsc-bio-ehr-es (Carrino et al., 2022)
CANTEMIST-NER	ES	PlanTL-GOB-ES/bsc-bio-ehr-es (Carrino et al., 2022)
DisTEMIST	ES	PlanTL-GOB-ES/bsc-bio-ehr-es (Carrino et al., 2022)
DrugTEMIST	ES	PlanTL-GOB-ES/bsc-bio-ehr-es (Carrino et al., 2022)
DrugTEMIST	EN	michiyasunaga/BioLinkBERT-base (Yasunaga et al., 2022)
DrugTEMIST	IT	IVN-RIN/bioBIT (Buonocore et al., 2023)

Unfortunately, we decided to participate on the MultiCardioNER shared task rather close to the deadline, meaning that we did not have time to employ a hyperparemeter search to optimize the models. Therefore, we used the same hyperparameters for all models, according to the best hyperparameters for fine-tuning the bsc-bio-ehr-es model on the CANTEMIST task: an effective batch size of 64 that was achieved by using an initial batch size of 32 with a gradient accumulation of 2 steps; a learning rate of 5e-5 which seemed to be the most consistent in terms of results during the fine-tuning experiments carried out by the authors; and finally, regarding the epochs, we evaluated each model at every epoch and kept the best one. These values are summed up in Table 3.7.

Table 3.7: The hyperparameter combination used for all models, with its basis on the experiments of Carrino et al. (2022).

Batch Size	Learning Rate	Epochs
64	$5e-5$	10

3.3.1.3 MultiCardioNER Experiments

Our participation in the MultiCardioNER shared task went beyond simply applying the chosen models to the dataset like we did for SympTEMIST and CANTEMIST. When devising an interesting plan with which to participate in this task with, we were intrigued by the fact that both the DisTEMIST and the DrugTEMIST Spanish corpora were derived from the same initial set of documents - the SPACCC corpus (Intxaurrondo & Krallinger, 2018) - with the only difference being the type of the entity mentions - diseases (ENFERMEDAD) and medications (FARMACO), respectively. Furthermore, we also knew that SympTEMIST was derived from the same corpus, annotating the symptom type entity mentions (SINTOMA). Then, upon further research, we noticed that there was another shared task that followed this trend of annotating over the SPACCC corpus with a single and unique type of entity - MedProcNER/ProcTEMIST (Lima-López, Farré-Maduell, Gascó, et al., 2023), that focused on medical procedures (PROCEDIMIENTO). With all of this in mind, our proposed methodology for MultiCardioNER was to experiment with fine-tuning three models: one solely with the entity type of the DisTEMIST dataset, another with the entity type of the DrugTEMIST dataset, and another one with all four entity types. That way, we could evaluate the benefit, or lack thereof, of having annotations regarding additional entity types.

To evaluate the proposed hypothesis, we conducted two experiments - the first experiment compared the model when fine-tuned on the combined dataset with the four entity types against the model when fine-tuned on just the DisTEMIST dataset; the second experiment also compared the model when fine-tuned on the combined dataset, but this time against the model when fine-tuned on just the DrugTEMIST dataset. Details of how we parsed the datasets will be laid out in the following section. This brings the total of runs submitted to four. Unfortunately, we could not contribute with submissions for the English and Italian subtasks in time.

In the following sections we will discuss the systematic approach we used to fine-tune and evaluate our models for Named Entity Recognition. First, the critical steps we took to convert the raw annotated data provided by the shared tasks, into a format suitable for fine-tuning our model with the Hugging Face libraries, are described. Following this, we provide the step-by-step process we employed for fine-tuning the models and evaluating our system.

3.3.1.4 Data Preprocessing

The original data for the Named Entity Recognition tasks, regardless of the shared task, is given in a collection of TXT and ANN files (standing for "text" and "annotation", respectively). The ANN files follow the BRAT annotation format, containing manually annotated information about the entities present in the TXT files, including their start and end positions, entity types, and the corresponding text spans. Listing 3.2 provides an example of a file of said format.

```
T1  ENFERMEDAD  40 55      diabetes tipo 2
T2  ENFERMEDAD  58 66      obesidad
T3  ENFERMEDAD  1086 1147  válvula mitral desestructurada y con prolapso de ambas valvas
T4  ENFERMEDAD  1735 1756  edema agudo de pulmón
T5  ENFERMEDAD  1870 1919  vegetaciones en ambas valvas de la válvula mitral
T6  ENFERMEDAD  1922 1949  insuficiencia mitral severa
T7  ENFERMEDAD  2118 2161  bloqueo aurículoventricular de tercer grado
T8  ENFERMEDAD  887 907    hemibloqueo anterior
T9  ENFERMEDAD  1817 1855  desestructuración de la válvula mitral
```

Listing 3.2: An example of an ANN file from the DisTEMIST corpus. It includes entity mentions of type ENFERMEDAD, as well as their mentions and spans inside the document.

We used Hugging Face's Datasets library as part of our data preprocessing step. This library is designed to support a wide range of file formats, enabling users to easily load, preprocess, and share datasets. However, while the BRAT format is highly informative and well-suited for manual annotation, it is not directly compatible with this library. Therefore, to ensure an efficient workflow throughout the entire pipeline - from parsing the dataset to loading the different splits - a critical step was converting these ANN files into a more standardized format that could be seamlessly processed: the CONLL format. Files that adhere to this format are composed by a series of paragraphs, each corresponding to an example (a sentence) in the original TXT files. Each line in this paragraph consequently corresponds to a token (word) from that example and contains information about that token, namely the actual token text, the original TXT file name to which that example corresponds to, the span of the token within that document, and a tag that follows the IOB2 tagging scheme mentioned earlier. This file type ensures that each token in the text is properly aligned with its corresponding entity label and allows seamless integration with Hugging Face. An example of this file type's structure is shown in Listing 3.3.

Next, we opted to standardize the size of the validation (dev) set relative to the train set for each task, rather than relying solely on the provided validation data (when available) or omitting the validation set entirely. Thus, we decided on an 80/20 split, where 80% of the total training plus validation examples would go to the resulting training split, and the remaining 20% would go to the validation split. The resulting splits and their statistics can be seen in Tables 3.8, 3.9 and 3.10, for SympTEMIST, CANTEMIST and MultiCardioNER respectively. The parsed datasets were made public in Hugging Face[3].

[3] https://huggingface.co/Rodrigo1771

```
En             casos_clinicos_cardiologia201  4133_4135  O
la             casos_clinicos_cardiologia201  4136_4138  O
unidad         casos_clinicos_cardiologia201  4139_4145  O
de             casos_clinicos_cardiologia201  4146_4148  O
críticos       casos_clinicos_cardiologia201  4149_4157  O
cardiológicos  casos_clinicos_cardiologia201  4158_4171  O
:              casos_clinicos_cardiologia201  4171_4172  O
Radiografía    casos_clinicos_cardiologia201  4173_4184  O
abdominal      casos_clinicos_cardiologia201  4185_4194  O
:              casos_clinicos_cardiologia201  4194_4195  O
niveles        casos_clinicos_cardiologia201  4196_4203  O
sugestivos     casos_clinicos_cardiologia201  4204_4214  O
de             casos_clinicos_cardiologia201  4215_4217  O
obstrucción    casos_clinicos_cardiologia201  4218_4229  B-ENFERMEDAD
intestinal     casos_clinicos_cardiologia201  4230_4240  I-ENFERMEDAD
alta           casos_clinicos_cardiologia201  4241_4245  I-ENFERMEDAD
en             casos_clinicos_cardiologia201  4246_4248  I-ENFERMEDAD
hipocondrio    casos_clinicos_cardiologia201  4249_4260  I-ENFERMEDAD
izquierdo      casos_clinicos_cardiologia201  4261_4270  I-ENFERMEDAD
.              casos_clinicos_cardiologia201  4270_4271  O

El             es-S1130-14732006000400007-1   3997_3999  O
aspecto        es-S1130-14732006000400007-1   4000_4007  O
estético       es-S1130-14732006000400007-1   4008_4016  O
de             es-S1130-14732006000400007-1   4017_4019  O
la             es-S1130-14732006000400007-1   4020_4022  O
herida         es-S1130-14732006000400007-1   4023_4029  B-ENFERMEDAD
quirúrgica     es-S1130-14732006000400007-1   4030_4040  I-ENFERMEDAD
es             es-S1130-14732006000400007-1   4041_4043  O
muy            es-S1130-14732006000400007-1   4044_4047  O
satisfactorio. es-S1130-14732006000400007-1   4048_4062  O
.              es-S1130-14732006000400007-1   4062_4063  O
```

Listing 3.3: An excerpt from the DisTEMIST CONLL training file. It includes two sentences, the names of the files they belong to, and the spans and tags of each token (word).

Table 3.8: Splits and statistics regarding the number of examples (sentences), and the number of unique and total entity mentions, for SympTEMIST-NER.

Dataset	Language	Split	Examples	Unique Mentions	Total Mentions
SympTEMIST-NER	ES	Train	9,597	5,020	7,267
	ES	Dev	2,519	1,454	1,822
	ES	Test	4,047	2,386	3,100

Table 3.9: Splits and statistics regarding the number of examples (sentences), and the number of unique and total entity mentions, for CANTEMIST-NER.

Dataset	Language	Split	Examples	Unique Mentions	Total Mentions
CANTEMIST-NER	ES	Train	29,030	2,960	9,739
	ES	Dev	7,354	952	2,537
	ES	Test	10,838	1,264	3,581

Table 3.10: Splits and statistics regarding the number of examples (sentences), and the number of unique and total entity mentions, for MultiCardioNER. The "Combined Dataset" combines the four entity types from DisTEMIST, DrugTEMIST, SympTEMIST and MedProcNER.

Dataset	Language	Split	Examples	Unique Mentions	Total Mentions
Combined Dataset	ES	Train	27,229	22,153	39,805
DisTEMIST	ES	Train	27,229	9,410	16,675
	ES	Dev	6,810	3,050	4,262
	ES	Test	14,614	4,038	7,868
DrugTEMIST	ES	Train	27,229	1,222	4,199
	ES	Dev	6,810	539	1,088
	ES	Test	14,614	509	1,716
	EN	Train	27,768	1,200	4,243
	EN	Dev	6,946	525	1,073
	EN	Test	14,715	518	1,721
	IT	Train	27,198	1,209	4,356
	IT	Dev	6,798	495	1,033
	IT	Test	14,605	519	1,786

3.3.1.5 Named Entity Recognition Pipeline

The Named Entity Recognition pipeline begins with fine-tuning each model on its respective dataset. We used the Hugging Face fine-tuning script[4] for this, slightly customizing it to our needs. The fine-tuning closely follows the aforementioned fine-tuning objective for the task of NER. Each model is trained with the hyperparameters shown in Table 3.7. Their performance is continuously monitored, once on each epoch, on the respective validation set. This allows us to gauge how well a model is learning and adjust the fine-tuning

[4] https://github.com/huggingface/transformers/blob/main/examples/pytorch/token-classification/run_ner.py

process accordingly. At the end of this process, the model from the epoch that yielded the best validation results (in terms of F1-score) is selected and the other ones discarded. This best-performing model is then evaluated on the test set to produce preliminary results, and finally uploaded to Hugging Face. Figure 3.1 illustrates the first part of the NER pipeline, i.e., the Hugging Face fine-tuning script.

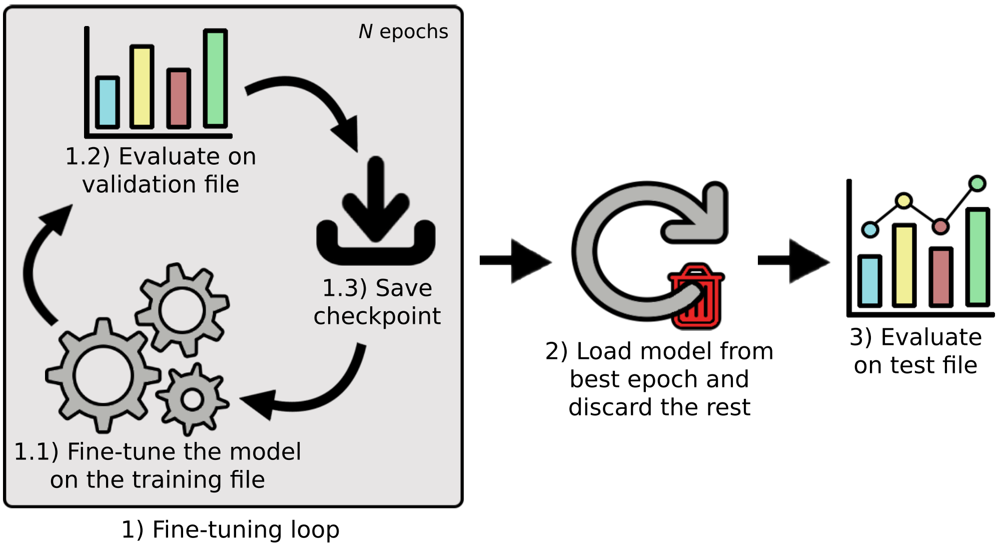

N epochs
1.2) Evaluate on validation file
1.3) Save checkpoint
1.1) Fine-tune the model on the training file
1) Fine-tuning loop
2) Load model from best epoch and discard the rest
3) Evaluate on test file

Figure 3.1: NER Pipeline Part 1 - The Hugging Face fine-tuning script.

It is important to note, though, that this preliminary evaluation, while useful for accessing the model's ability, did not fully align with the official evaluation methods used by the shared tasks. This disparity stems from the differences in how the datasets were processed and presented to the model: the fine-tuning script evaluated the model sentence by sentence, using the CONLL files we had previously prepared in order to use Hugging Face's Datasets library. In this setup, each sentence was treated independently, with entity mention spans relative to the beginning of each sentence, not the entire document. This led to metrics that were sentence-based, instead of document-based like the ones that the official evaluation libraries provide, which we thought affected the reliability of those metrics when comparing our approach with others. For this reason, we decided to use the official evaluation libraries in order to later provide comparisons that are as trustworthy as possible.

However, we ran into a significant problem when evaluating documents as a whole, which was that the context window of the BERT-based models is limited to 512 tokens. Consequently, documents were truncated when they exceeded this length, meaning that the models never had the opportunity to identify mentions that appeared beyond this token limit. This severely impacted recall: although the models could label the mentions that appeared within the 512-token window with good and even great precision, they would inevitably miss a substantial number of the total mentions - the ones that appeared in the portions of the documents that were cut off. This imbalance between precision and recall led to poor F1-scores. Unfortunately, despite mentioning it in our submission paper (Gonçalves & Lamúrias, 2024) as a possible reason for the poor results, this issue was confirmed too late to rectify before the official results for the MultiCardioNER shared task were locked, leading to reported metrics that didn't fully reflect the capabilities of our approach.

Nevertheless, we still wanted to address this for this dissertation and so we implemented a simple post-training processing step right before obtaining the predictions to use with the official evaluation library. Once the models were saved on our Hugging Face hub, we revisited the TXT files of the test sets, breaking them down sentence by sentence into JSON files. Each JSON file contained essential information for reconstructing the document-level predictions for each subtask: the sentence itself, the name of the original document, and the offset of the sentence within that document. This allowed us to adjust the spans of the predicted entity mentions based on their original position within the document and accurately predict over each full document without directly feeding it into the model, thus bypassing the context length limitations.

Finally, after obtaining these document-level predictions, a series of post-processing steps were applied, for instance joining tokens that had been split during the model's prediction phase (i.e. combining labels with the prefixes "B-" and "I-" back into a single, cohesive label). These refined predictions were then saved in a TSV file format, which was compatible with the official evaluation libraries of each shared task. Feeding these TSV files into the shared tasks' evaluation libraries allowed us to obtain the final metrics, which more accurately reflected the model's performance and provided a more reliable basis of comparison. Figure 3.2 illustrates the second part of the NER fine-tuning pipeline, i.e., this process of obtaining the results.

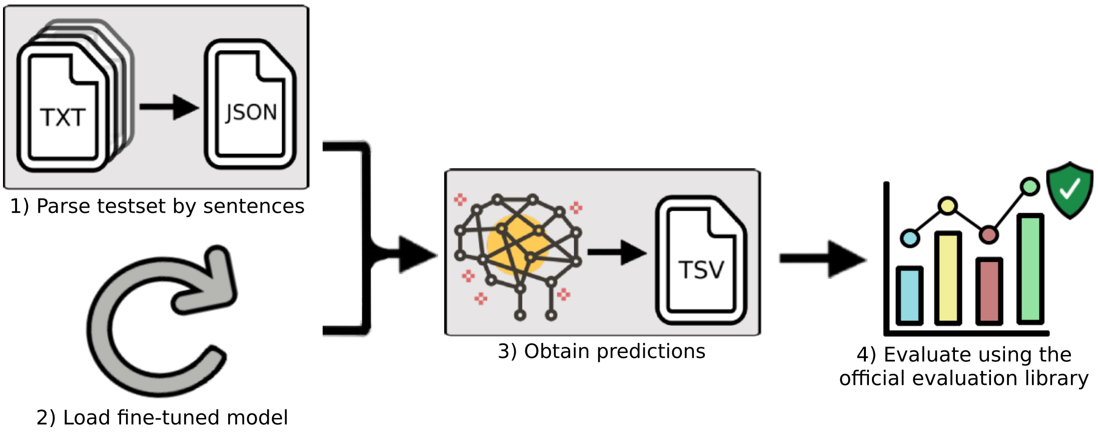

TXT
JSON
1) Parse testset by sentences
2) Load fine-tuned model
TSV
3) Obtain predictions
4) Evaluate using the official evaluation library

Figure 3.2: NER Pipeline Part 2 - Obtaining the results of the fine-tuned model.

3.3.2 Data Augmentation

3.3.2.1 Word Embeddings

The most straightforward way to implement Data Augmentation for NLP tasks is to replace entity mentions with their synonyms. This aims to improve the model's ability to generalize to new and unseen mentions by expanding the dataset with more diverse, yet semantically similar training data. To obtain these synonyms, we chose to use word embeddings (Bengio et al., 2003) as they represent an extremely efficient technique for capturing semantic links between words, especially compared to Large Language Models (LLMs) which are computationally expensive and over-parameterized for simple synonym replacement tasks.

Word embeddings are vector representations of words that encode linguistic context. Therefore, words with comparable meanings will have similar vectors, maintaining semantic closeness. This makes them a strong candidate for discovering appropriate substitutes for concepts without significantly affecting the sentence's meaning, and ensures that the generated data remains relevant to the task. There are several types of word embedding models, each designed to capture semantic relationships in different ways.

Word2Vec (Mikolov et al., 2013), for instance, comes in two architectures - Skip-gram and Continuous Bag of Words (CBOW) - and is designed to capture semantic relationships between words by predicting either the surrounding words based on a target word (Skip-gram) or a target word based on surrounding words (CBOW). Figure 3.3 illustrates these two architectures.

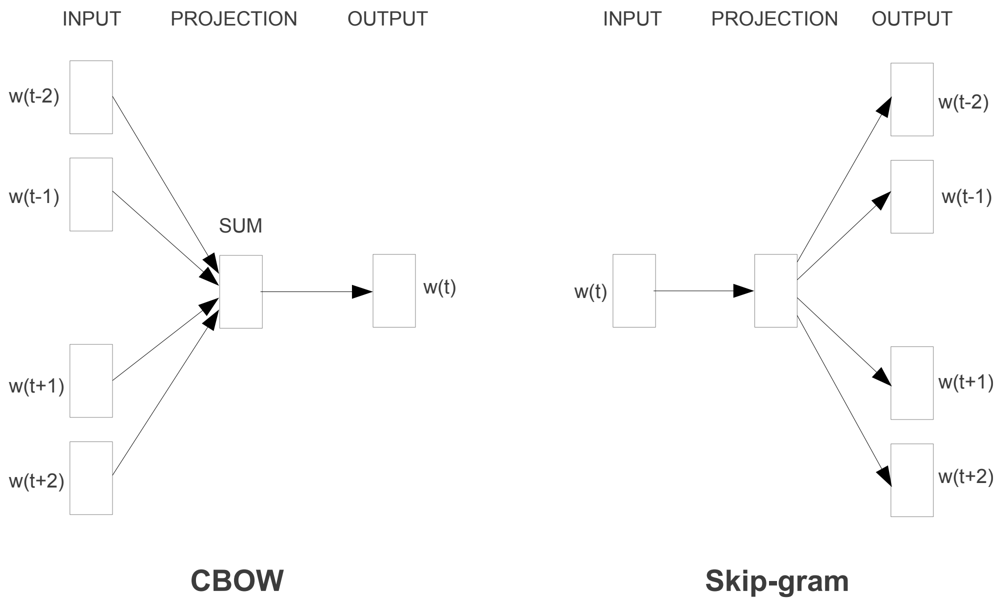


Figure 3.3: The CBOW and Skip-gram architectures. Adapted from Mikolov et al., 2013.

Another example is FastText (Bojanowski et al., 2017), which extends Word2Vec by incorporating subword information, breaking words into character n-grams. This allows FastText to handle rare and out-of-vocabulary words more effectively, which is particularly useful when dealing with domain-specific datasets. We decided to include both types of models in this dissertation, in order to compare their effectiveness in our data augmentation experiments.

The FastText project provides robust pre-trained models for lots of languages, including the ones we focused on (Grave et al., 2018). They were trained on Common Crawl[5] and Wikipedia[6], using CBOW with position-weights, with a vocabulary of 2,000,000 words and word vector dimension of 300. Other hyperparameters include n-grams of length 5, a window of size 5, and 10 negatives, as described by Grave et al., 2018.

However, we were not able to attain access to any pre-trained Word2Vec models. Therefore, we decided to train our own. For this, we used the wordvectors[7] project, which provides a pipeline for training Word2Vec models. We trained five such models, each corresponding to a different language - Spanish, English, French, Italian, and Portuguese. These five languages cover every NER and EL subtask. For a given language, the pipeline first builds a corpus comprised of every sentence from the Wikimedia database backup dump[8] from that language, before training a Word2Vec model using the Gensim library (Řehůřek & Sojka, 2010), a popular open-source library for training word embeddings and finding synonyms for text augmentation. Table 3.11 presents the size of each corpus in number of characters, words and sentences.

Table 3.11: The number of characters, words and sentences of each language corpus. The English corpus is noticeably larger than the other ones, which is to be expected when considering resource availability in English against other languages.

Language	Characters	Words	Sentences
ES	5,203,677,629	874,854,203	35,032,989
EN	21,723,738,961	3,624,113,529	167,572,384
FR	6,555,040,563	1,073,343,257	47,125,581
IT	4,128,591,527	656,653,213	26,613,359
PT	2,339,733,765	389,536,649	16,827,129

We also used the Skip-gram architecture in the Word2Vec models as it is especially effective at capturing semantic nuances, in addition to being generally better for infrequent words as it provides more training opportunities for them (given its nature of prioritizing predicting context words for a given target word). This makes it well suited for the kind of synonym replacement needed in domain-specific NER and EL tasks. Other

[5] http://commoncrawl.org/

[6] https://www.wikipedia.org

[7] https://github.com/Kyubyong/wordvectors

[8] https://dumps.wikimedia.org

hyperparameters include a window size of 10, a negative sampling value of 5 and a vector size of 300, all standard values for training this type of model with a relatively large dataset (Kulshrestha, 2021). We also capped the vocabulary size of the models at around 500,000 as we felt that was big enough without compromising efficiency.

3.3.2.2 Augmentation Algorithm

With the word embeddings obtained, we now focused on expanding the training files for each NER subtask with diverse yet semantically consistent variations of each entity in the corpus. This was achieved by replacing words in the entity mentions, at random, with their closest synonyms, as determined by the word embeddings obtained in the previous step. The augmentation algorithm operates on a sentence-by-sentence basis, leveraging the CONLL file format in which the data is stored (Listing 3.3).

For each sentence in the dataset, the augmentation is performed three times. This modest "augmentation factor" of three allows for greater variety in the dataset by creating multiple versions of each sentence with different entity representations instead of just one, without inflating the dataset with countless versions of the same sentence to the point of compromising computational efficiency. The algorithm proceeds by first identifying the entity mentions in the sentence and selecting one word from each mention (excluding structure words) to replace with a synonym. It ensures that no word within the mention is replaced more than once, increasing the variety of the augmented dataset. However, in cases where the entity mention contains fewer words than the augmentation factor value, every word is bound to be replaced before the augmentation factor is exhausted. When this happens, the algorithm will continue by replacing words in that mention again, but now using their second-closest synonym. This approach guarantees that even short mentions are augmented multiple times. Finally, if no synonyms are found for any mention in the sentence, then no augmentation is performed on that sentence. Listing 3.4 presents the algorithm's pseudocode.

A key factor in selecting synonyms is the use of a similarity threshold. Two words are considered synonyms only if their similarity score is higher than this predefined threshold, ensuring that the replacement word is semantically close to the original. We chose {0.75, 0.80, 0.85, 0.90}[9] as our similarity threshold search space, which resulted in four augmented training files per subtask. To illustrate the impact of the similarity threshold, Tables 3.12 and 3.13 present the number of training examples (sentences), unique and total mentions from the training files augmented using Word2Vec and the training files augmented using FastText, respectively, given the similarity thresholds. Furthermore, Table 3.14 provides an augmentation example from the DisTEMIST training file, using both Word2Vec and FastText. We can observe that FastText, with its subword knowledge, produced replacements that were variants of the original term. Word2Vec,

[9] With similarity thresholds smaller than 0.75, the generated synonyms deviated too much from the original meaning.

on the other hand, generated more diverse mentions, and despite representing different diseases entirely, they still fall under the same type "disease" nonetheless, which overall increases the variety of the dataset.

```
// choose_random(words) never returns the same word twice
// count_replacements(word) tracks how many times a word has been replaced
// find_synonyms(word, embeddings, threshold) returns all synonyms, ordered by similarity

Require dataset D, embeddings E, augmentation factor A, similarity thresholds T

function get_synonym(mention, embeddings, threshold, current_augmentation_factor)
    words_to_replace ← exclude_structure_words(mention)

    for each word w ∈ choose_random(words_to_replace) do
        replacements_count ← count_replacements(w)
        if replacements_count < current_augmentation_factor then
            synonyms ← find_synonyms(w, embeddings, threshold)
                        if synonyms exists then
                return synonyms[replacements_count]
            end if
        end if
    end for

    return ∅
end function

augmented_sentences ← ∅
for each sentence s ∈ D do
    augmented_sentences ← augmented_sentences ∪ {s}
    mentions ← get_mentions(s)

    for i = 1 to A do
        for each mention m ∈ mentions do
            synonym ← get_synonym(m, E, T, i)

            if synonym exists then
                s_aug ← replace_with_synonym(s, synonym, m)
            end if
        end for

        if s_aug exists then
            augmented_sentences ← augmented_sentences ∪ {s_aug}
        end if
    end for
end for

return augmented_sentences
```

Listing 3.4: The pseudocode for the NER data augmentation algorithm.

Table 3.12: The number of training examples (sentences), unique and total mentions from the NER training files augmented using Word2Vec, given the similarity thresholds.

Dataset	Language	Similarity Threshold	Examples	Unique Mentions	Total Mentions
SympTEMIST-NER	ES	0.75	15,848	9,728	20,178
		0.80	13,389	7,567	15,979
		0.85	11,133	5,926	11,393
		0.90	9,845	5,193	8,109
CANTEMIST-NER	ES	0.75	43,811	6,642	31,404
		0.80	38,739	5,383	24,558
		0.85	32,675	4,071	15,656
		0.90	29,083	2,996	9,849
DisTEMIST	ES	0.75	46,093	20,748	48,761
		0.80	39,567	16,162	39,230
		0.85	31,947	11,718	26,329
		0.90	28,231	9,945	18,718
DrugTEMIST	ES	0.75	31,229	1,827	14,046
		0.80	30,970	1,724	13,434
		0.85	29,797	1,516	11,373
		0.90	27,682	1,270	5,905
	EN	0.75	32,232	1,872	14,416
		0.80	31,536	1,699	13,259
		0.85	29,253	1,384	8,421
		0.90	27,967	1,254	4,787
	IT	0.75	31,901	1,890	15,058
		0.80	31,619	1,781	14,647
		0.85	30,642	1,610	13,191
		0.90	27,715	1,294	6,412

Table 3.13: The number of training examples (sentences), unique and total mentions from the NER training files augmented using FastText, given the similarity thresholds.

Dataset	Language	Similarity Threshold	Examples	Unique Mentions	Total Mentions
SympTEMIST-NER	ES	0.75	16,483	11,477	20,834
		0.80	13,013	8,079	14,621
		0.85	10,936	6,219	10,453
		0.90	9,929	5,372	8,141
CANTEMIST-NER	ES	0.75	39,426	6,829	25,766
		0.80	34,619	4,996	18,776
		0.85	31,215	3,847	13,368
		0.90	29,299	3,162	10,185
DisTEMIST	ES	0.75	44,938	21,562	46,788
		0.80	35,750	14,807	32,593
		0.85	30,971	11,840	23,829
		0.90	28,539	10,144	19,201
DrugTEMIST	ES	0.75	31,229	2,261	13,844
		0.80	29,959	1,877	11,450
		0.85	28,504	1,538	8,203
		0.90	27,369	1,305	4,937
	EN	0.75	32,447	2,286	14,768
		0.80	30,812	1,865	11,764
		0.85	28,668	1,402	6,834
		0.90	27,841	1,235	4,469
	IT	0.75	31,053	2,163	13,821
		0.80	30,064	1,853	11,990
		0.85	28,875	1,549	9,564
		0.90	27,796	1,317	6,652

Table 3.14: Examples of augmented sentences for NER, from the DisTEMIST training file, using both Word2Vec and FastText with a similarity threshold of 0.75. Blue text indicates entity mention replacement.

Version	Sentence
Original	... lo que permite tan solo una apertura interincisal máxima de 4 mm además de una maloclusión de clase II.
Word2Vec	... lo que permite tan solo una apertura interincisal máxima de 4 mm además de una plagiocefalia de clase II.
	... lo que permite tan solo una apertura interincisal máxima de 4 mm además de una encefalocele de clase II.
	... lo que permite tan solo una apertura interincisal máxima de 4 mm además de una macroglosia de clase II.
FastText	... lo que permite tan solo una apertura interincisal máxima de 4 mm además de una maloclusiones de clase II.
	... lo que permite tan solo una apertura interincisal máxima de 4 mm además de una Maloclusión de clase II.

We then proceeded with training the NER models as before, but now with the new augmented datasets. For the English and Italian DrugTEMIST datasets, we used the four datasets that correspond to the four predefined similarity thresholds. However, for the Spanish tasks, we first evaluated the performance of the four augmented SympTEMIST-NER datasets to identify the optimal similarity threshold, and then applied that threshold to the remaining Spanish datasets, in order to maintain consistency across languages, while also reducing computational overhead.

3.4 Entity Linking Experiments

Having covered the Named Entity Recognition experiments in depth, this section now focuses on Entity Linking. As mentioned before, by linking identified mentions to relevant items in a knowledge base, Entity Linking contributes to a greater level of semantic context and comprehension. This subsection is subdivided into the same two sections as the previous one, first exploring our transfer learning approach to EL from the use of the SapBERT model and framework, to the hyperparameter search intended to maximize performance, the data preprocessing and finally the whole pipeline. Then, we will also discuss the data augmentation attempt at improving the performance of our initial EL transfer learning experiments.

3.4.1 Transfer Learning

3.4.1.1 The SapBERT Model And Framework

SapBERT, or Self-aligning pre-trained BERT (Liu, Shareghi, et al., 2021), is both a transformer-based model and an overall training framework specifically designed to enhance the models' internal representations of biomedical entities. It achieves this by leveraging a technique called Self-Alignment Pre-training (SAP), which is a simple pre-training objective that clusters synonyms of the same concept, effectively making the embedding space much more coherent. Two different SapBERT models are available, an English only model, using BiomedBERT (Gu et al., 2022) as the base model, and a multilingual model (Liu, Vulić, et al., 2021), which was built on top of XLM-RoBERTa (Conneau et al., 2020).

The English models' pre-training process starts by first leveraging UMLS (Bodenreider, 2004), which contains a comprehensive collection of 4M+ biomedical concepts and 10M+ synonyms from over 150 controlled vocabularies. This provides an extensive set of (*name*, *code*) pairs that map medical entity mentions (*name*) to their UMLS identifiers (*code*). Next, a technique is used to find the (*name*, *code*) pairs that are either not close enough or too close in the embedding space, as those pairs are the ones that should be corrected in order to improve the models' representations. This is done by random batching those pairs: every single one of the pairs within a given mini-batch is used to build all possible triplets of type (x_a, x_p, x_n), where x_a is the anchor entity mention, x_p a positive match of x_a (i.e. they have the same $code$) and x_n a negative match of x_a (i.e. they have a different $code$). Following that, they filter for the triplets with the negative sample closer to the positive sample by a certain margin λ, effectively collecting the hardest mentions to classify as negative or positive given the anchors, and subsequently, the ones that should be rectified. Every one of the filtered triplets contributes one positive pair (x_a, x_p) and one negative pair (x_a, x_n). However, because random batching on the original (*name*, *code*) list can lead to very few (if not none) positive pairs within a mini-batch, all possible positive pairs are generated beforehand by traversing all pairwise combinations of names with the same code and forming triplets of type (x_1, x_2, *code*). These triplets make up the actual data that is fed to the model, which is then deconstructed during the SAP procedure to form the (x_a, x_p, x_n) triplets. This process offers a lot more balance between positive and negative pairs, by getting rid of a lot of the uninteresting negative pairs. Figure 3.4 provides a short example of this process (up until the loss function).

Finally, an adaptation of the Multi-Similarity Loss function (Wang et al., 2019) is used to update the model's parameters:

$$\mathcal{L} = \frac{1}{|\mathcal{X}_b|} \sum_{i=1}^{|\mathcal{X}_b|} \left[\frac{1}{\alpha} \log \left(1 + \sum_{n \in \mathcal{N}_i} e^{\alpha(\mathcal{S}_{in} - \epsilon)} \right) + \frac{1}{\beta} \log \left(1 + \sum_{p \in \mathcal{P}_i} e^{-\beta(\mathcal{S}_{ip} - \epsilon)} \right) \right]$$

First, the pairwise cosine similarity of all the BERT-produced *name* representations is computed and a similarity matrix $\mathcal{S} \in \mathbb{R}^{|\mathcal{X}_b| \times |\mathcal{X}_b|}$ is obtained, where each entry $\mathcal{S}_{ij}$

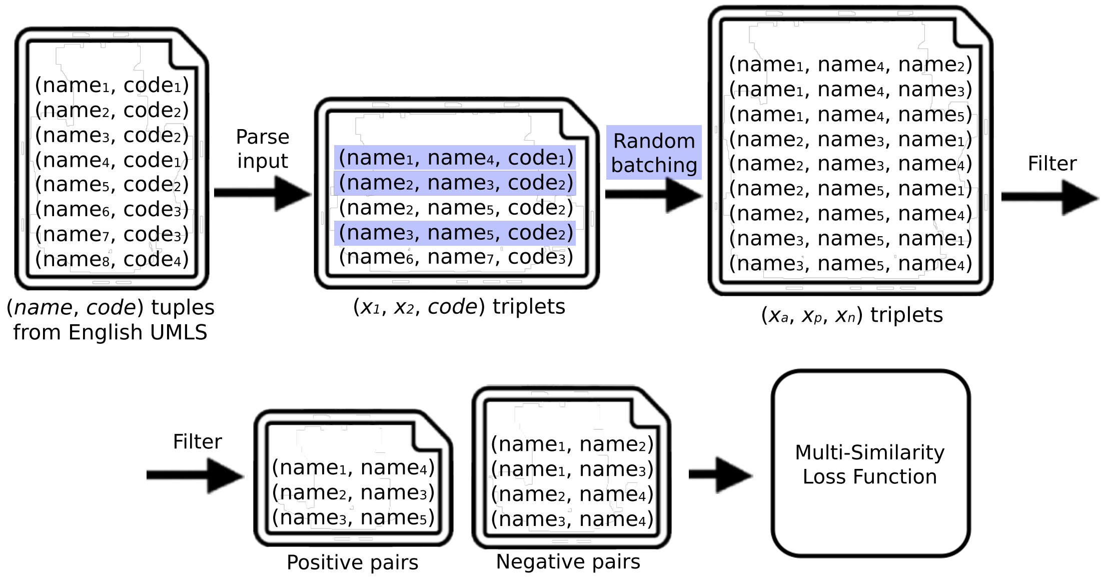


Figure 3.4: A small example of the Self-Alignment Pre-training procedure, from the UMLS data to the loss function.

corresponds to the cosine similarity between the i-th and j-th names in mini-batch b. Then, the loss function above is applied to that batch, where α, β are temperature scales; ϵ is an offset applied on the similarity matrix; and $\mathcal{P}_i$, $\mathcal{N}_i$ are indices of positive and negative samples of the anchor i. While the first term in the equation pushes negative pairs away from each other, the second term pulls positive pairs together. This dynamic allows for a recalibration of the embedding space, leveraging similarities and dissimilarities between positive and negative pairs respectively to re-weight the importance of the samples.

The multilingual model is an extension of the original English SapBERT procedure that attempts to solve the issue of expert knowledge only being abundantly available for a handful of languages, by transferring domain-specific knowledge from resource-rich languages to resource-poor ones. To do this, a multilingual extension of the SAP technique was proposed: first, instead of only considering English UMLS, other UMLS languages are taken into account for the self-alignment process. The idea is that mentions with the same code should all have similar representations regardless of their language. For instance, vaccination (EN), active immunization (EN) and vacinación (ES) all share the same code (C0042196); thus, they should all have similar representations. To achieve this, the (x_1, x_2, *code*) triplets are built in the same way as before, but with mentions (names) of all the considered languages, not just English. The second improvement over the original SapBERT involves combining domain-specific synonyms with general-domain translation data. For this end, translation pairs of type (x_i, x_j), where x_i and x_j represent the same word/sentence, but in different languages (translated from one to the other), are assigned a pseudo-code $code_{ij}$ (usually generated by concatenating x_i and x_j), in order to form two new pairs, (x_i, $code_{ij}$) and (x_j, $code_{ij}$), that have the same structure as the previous (*name*, *code*) pairs. These new pairs can then be fed to the beginning of the self-alignment pipeline to form (x_1, x_2, *code*) triplets - in this case (x_i, x_j, $code_{ij}$) triplets - just like what happens with the (*name*, *code*) pairs. This will result in additional triplets that contain general-domain translation, not biomedical knowledge. What results is a model that can capture the intricate biomedical knowledge from resource-rich languages like English and effectively transfer it to resource-poor languages, thereby being closer to bridging the multilingual gap in the biomedical domain.

The fine-tuning approach for both variants is very similar to the original self-alignment pre-training objective: the data is parsed into the aforementioned (x_1, x_2, *code*) triplets and fed into the SAP pipeline as normal.

At evaluation time, the model needs to link an entity mention to a code from a designated knowledge base. To achieve this, a dictionary with mappings of biomedical concepts to the knowledge base codes is used, and a simple Nearest Neighbour Search (NSS) between the queried mention and the mentions in the dictionary is sufficient for making a prediction. If evaluating based on the Accuracy@1, the code that corresponds to the top-ranked mention from this NSS is the prediction output, obtained by a simple dictionary look up. For the Accuracy@N, the N top-ranked mentions are considered.

Regarding the results reported by the authors, the SapBERT training procedure consistently improves all 7 tested BERT-based models across all 6 english datasets used for evaluating this approach. Furthermore, the SAP-enhanced models achieved new state-of-the-art with statistical significance on 5 of those 6 datasets, only needing the pre-training step to do this against otherwise fine-tuning approaches. On the other hand, multilingual SapBERT's results show that applying this pipeline to multilingual models like mBERT (Devlin et al., 2019) and XLM-Roberta (Conneau et al., 2020) leads to staggering relative gains, with its application to monolingual BERTs for each tested language also yielding substantial gains across all languages.

The impressive results, combined with the relative simplicity of the approach, motivate our choice for this pipeline in our experiments - it should enable us to benefit from the advantages of self-alignment, not in pre-training but in our fine-tuning of different models, including but not limited to the actual pre-trained SapBERT model.

3.4.1.2 Pre-trained LLMs Chosen and Hyperparameter Search

After choosing the framework with which to tackle the task of Entity Linking with, we concentrated on analysing the state-of-the-art landscape of monolingual models for the languages that we decided to focus on, before devising comprehensive hyperparemeter experiments to try and get the most out of the models. This approach allowed us to fine-tune the models for optimal performance considering our resources and everything we used.

SympTEMIST, the Entity Linking related shared task that we focused our efforts on, has two subtasks: through the second subtask SymptomNorm, it supports the Spanish language; and through subtask three, SymptomMultiNorm, it offers support for many languages, including English, French, Italian and Portuguese, the languages we chose to explore together with Spanish. Table 3.15 lists the models we chose for each language. As far as our research showed us, they are the state-of-the-art monolingual large decoder-based language models from their respective languages.

Table 3.15: The models (their Hugging Face repository IDs) chosen for the Entity Linking experiments, per language. The EL experiments were all performed on SympTEMIST.

Language	Model
Multilingual	cambridgeltl/SapBERT-UMLS-2020AB-all-lang-from-XLMR (Liu, Vulić, et al., 2021)
ES	PlanTL-GOB-ES/bsc-bio-ehr-es (Carrino et al., 2022)
EN	emilyalsentzer/Bio_ClinicalBERT (Alsentzer et al., 2019)
EN	michiyasunaga/BioLinkBERT-base (Yasunaga et al., 2022)
EN	microsoft/BiomedNLP-BiomedBERT-base-uncased-abstract-fulltext (Gu et al., 2022)
FR	almanach/camembert-bio-base (L. Martin et al., 2020)
FR	quinten-datalab/AliBERT-7GB (Berhe et al., 2023)
FR	Dr-BERT/DrBERT-7GB (Labrak et al., 2023)
IT	IVN-RIN/bioBIT (Buonocore et al., 2023)
IT	IVN-RIN/medBIT (Buonocore et al., 2023)
IT	IVN-RIN/MedPsyNIT (Buonocore et al., 2023)
PT	pucpr/biobertpt-all (Schneider et al., 2020)
PT	pucpr/biobertpt-clin (Schneider et al., 2020)
PT	neuralmind/bert-base-portuguese-cased (Souza et al., 2020a)
PT	PORTULAN/albertina-100m-portuguese-ptpt-encoder (Santos et al., 2024)

We also chose to implement a hyperparameter grid search. For this, we evaluated each model on a validation set made up of a small percentage of examples from the training data, which will be elaborated on in the subsequent sections. We focused on three key hyperparameters: the batch size, the learning rate, and the number of epochs. Batch size refers to the number of training examples utilized in one iteration, with larger batch sizes generally providing more stable gradient estimates but requiring more memory (this was a limiting factor for us, as we could only go up to a batch size of 512 due to the available computational resources). Learning rate is the step size at each iteration while moving towards a minimum of the loss function, where a higher learning rate can speed up training but might overshoot the optimal solution, and a lower learning rate provides more precise updates but can significantly slow down the training process. The number of epochs is the number of complete passes through the entire training dataset.

We evaluated each model at every epoch to assess the best one.

By systematically varying these hyperparameters within a defined range, we aimed to identify the combination that yielded the best performance on our validation set. Specifically, we tested a range of values for each hyperparameter, highlighted in Table 3.17. These values were selected based on their significance in the fine-tuning process as outlined in the SapBERT papers (Liu, Shareghi, et al., 2021; Liu, Vulić, et al., 2021). This approach allowed us to observe the effects of different configurations and select the most effective setup for each model, which are shown in Table 3.16.

Table 3.16: The best hyperparameter combination for each model.

Dataset's Language	Model	Batch Size	Learning Rate	Epochs
ES	cambridgeltl/SapBERT-UMLS-2020AB-all-lang-from-XLMR	64	$1e-5$	3
ES	PlanTL-GOB-ES/bsc-bio-ehr-es	128	$1e-4$	20
EN	cambridgeltl/SapBERT-UMLS-2020AB-all-lang-from-XLMR	64	$2e-5$	2
EN	emilyalsentzer/Bio_ClinicalBERT	64	$2e-5$	10
EN	michiyasunaga/BioLinkBERT-base	128	$2e-5$	17
EN	microsoft/BiomedNLP-BiomedBERT-base-uncased-abstract-fulltext	256	$5e-5$	15
FR	cambridgeltl/SapBERT-UMLS-2020AB-all-lang-from-XLMR	64	$5e-5$	3
FR	almanach/camembert-bio-base	64	$1e-5$	20
FR	quinten-datalab/AliBERT-7GB	64	$1e-4$	2
FR	Dr-BERT/DrBERT-7GB	64	$1e-5$	2
IT	cambridgeltl/SapBERT-UMLS-2020AB-all-lang-from-XLMR	64	$5e-5$	5
IT	IVN-RIN/bioBIT	64	$5e-5$	5
IT	IVN-RIN/medBIT	256	$1e-4$	17
IT	IVN-RIN/MedPsyNIT	64	$1e-4$	14
PT	cambridgeltl/SapBERT-UMLS-2020AB-all-lang-from-XLMR	128	$1e-4$	1
PT	pucpr/biobertpt-all	64	$5e-5$	8
PT	pucpr/biobertpt-clin	256	$2e-5$	13
PT	neuralmind/bert-base-portuguese-cased	64	$5e-5$	3
PT	PORTULAN/albertina-100m-portuguese-ptpt-encoder	64	$2e-5$	14

Table 3.17: The search space for each hyperparameter. These values were selected based on their significance in the fine-tuning process as outlined in the SapBERT papers (Liu, Shareghi, et al., 2021; Liu, Vulić, et al., 2021).

Hyperparameter	Search Space
Batch Size	{64, 128, 256, 512}
Learning Rate	$\{1e-4, 5e-5, 2e-5, 1e-5\}$
Epochs	20

3.4.1.3 Data Preprocessing

We undertook several crucial data preprocessing steps to prepare the raw data for being fed to the models. The original train and test sets, provided in TSV (Tab Separated Values) format, were transformed into training and test files with a more suitable format for our tasks. Specifically, we converted those original TSV files into a text file format where each line follows the $(x1, x2, code)$ structure explored before. This format is essential for aligning the data with the requirements of the SapBERT fine-tuning procedure. Furthermore, due to this input format, entity mentions that do not have synonyms present in the original training data are excluded from the training and test files by this preprocessing step, focusing our dataset on the relevant mention pairs.

We further refined the dataset for the hyperparameter search phase by splitting the training data into two subsets: 80% of the entity mentions for training, and the remaining 20% for validation. This division allowed us to fine-tune the hyperparameters effectively and assess their impact on model performance. Then, we used the complete training and test files for fine-tuning and evaluating the final model with the optimal hyperparameters, ensuring that the evaluation was conducted on the most comprehensive dataset. Table 3.18 presents some statistics regarding the number of unique and total entity mentions and codes, as well as total training synonyms, for each dataset. Additionally, Figure 3.5 provides further clarity on the final structure of the preprocessed files.

In addition to the processing of the main dataset, we also created the dictionary files, crucial for the Entity Linking task. The gazetteer provided by the organizers was initially in TSV format and was converted into a text file with the format $(code, x)$, where $code$ is the code for entity mention x. This conversion enabled us to efficiently map codes to their respective mentions during the linking process. However, this gazetteer was only provided in Spanish. We suspect this is due to the fact that the experimental multilingual EL subtask was primarily meant for examining automatic translation strategies, and the translation of the gazetteer would be part of the challenge. Nevertheless, we compromised by creating a different type of dictionary for each of the remaining languages, extracted from their respective training files (either from the 80% or from the whole training data, depending if performing hyperparameter search or training the final models, respectively) - and thus containing drastically fewer entries. We also included the mentions that have no synonyms, and are discarded when parsing the train set, in both types of dictionaries, as they could still help in evaluation. Figure 3.6 illustrates the preprocessing of each dictionary.

Table 3.18: The number of unique entity mentions and codes, as well as total training synonyms, for each SympTEMIST (sub)dataset. The Train and Test splits refer to the files used to fine-tune the final models with the best hyperparameter combination, not the files used to perform the hyperparameter search. The training splits have no "Total Mentions" because in this approach of matching entity mentions with their synonyms in the training file, we filtered for duplicates so as to not have duplicate synonyms, meaning that all mentions are unique. Likewise, the test splits don't have "Total Synonyms", because we only pair synonyms when training the model, not at the evaluation step.

Dataset	Language	Split	Unique Codes	Unique Mentions	Total Mentions	Total Synonyms
SympTEMIST-EL	ES	Train	437	1,341	-	2,377
	ES	Test	1,391	2,144	2,848	-
SympTEMIST-MEL	EN	Train	162	414	-	414
	EN	Test	704	903	1,600	-
	FR	Train	143	365	-	398
	FR	Test	668	856	1,425	-
	IT	Train	150	384	-	403
	IT	Test	708	905	1,544	-
	PT	Train	127	329	-	352
	PT	Test	690	909	1,521	-

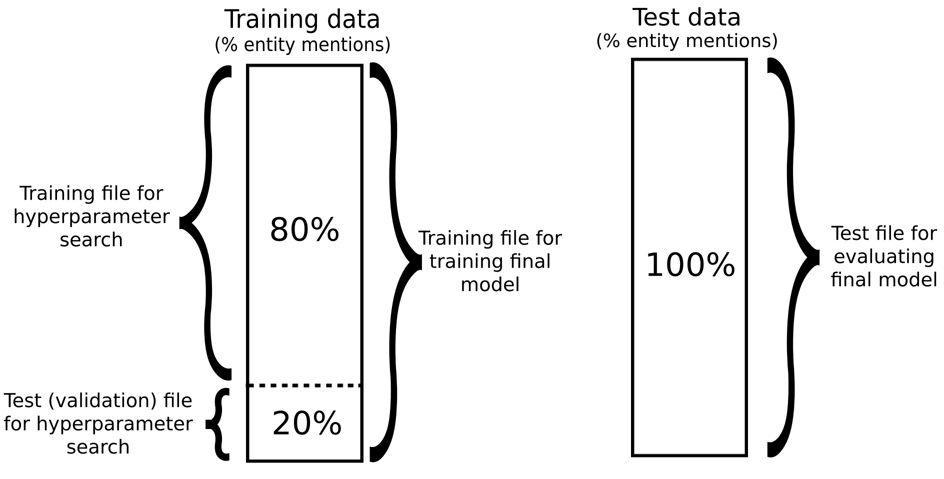

Training data
(% entity mentions)
Test data
(% entity mentions)
Training file for hyperparameter search
80%
Training file for training final model
100%
Test file for evaluating final model
Test (validation) file for hyperparameter search
20%

Figure 3.5: The resulting training and test files for EL.

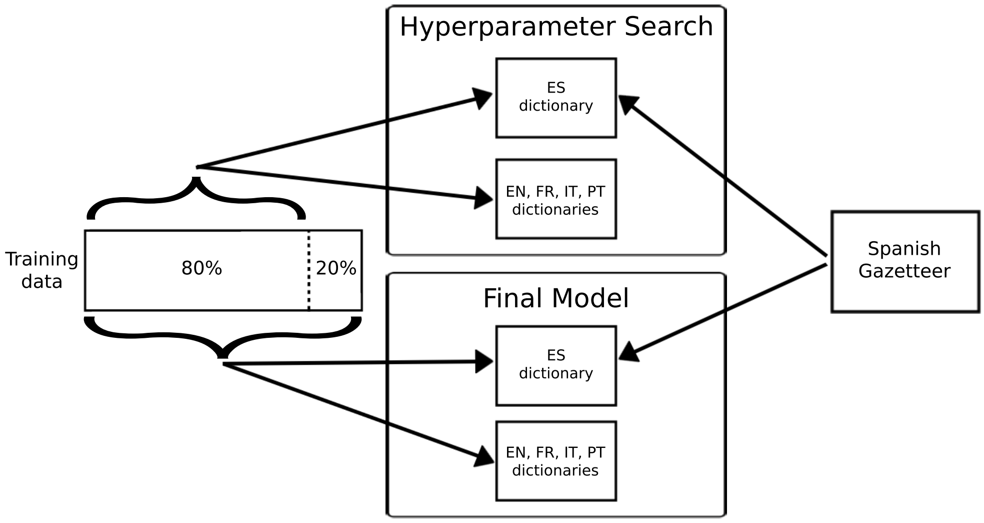


Figure 3.6: The resulting dictionary files for EL.

Additionally, Table 3.19 contains the results of some experiments we conducted on the validation set of the Spanish subtrack, in order to confirm that the chosen compromise for the languages with no gazetteer would offer comparable results to the full dictionary. In these experiments, we compared 1) the full Spanish dictionary, 2) a dictionary made up of only the Spanish training file, and 3) a dictionary made up of only the Spanish training file, and augmented with the entity mentions with no synonyms that were discarded during the parsing of the training file. The experiments were performed with the same model (cambridgeltl/SapBERT-UMLS-2020AB-all-lang-from-XLMR), the same hyperparameters (512 batch size; $2e-5$ learning rate; 20 epochs), and the same training and test files (the training and validation files used for the hyperparameter search). We can conclude that the dictionary composed by the training file, and augmented with the synonymless mentions, provides the closest performance.

Table 3.19: Dictionary experiments. The dictionary composed by the training file, and augmented with the synonymless entity mentions, provides the closest performance to the full Spanish dictionary.

Dictionary	Accuracy
Spanish Dictionary	0.7016
Training file	0.4362
Training file with synonymless mentions*	0.6126

3.4.1.4 Entity Linking Pipeline

The process of fine-tuning and evaluating the models was fairly straight-forward. We focused on two main stages: the hyperparameter search, and final model fine-tuning. For each language, we first performed a hyperparameter grid search: the training script iterates over the hyperparameter search spaces and models previously presented, and each models' performance, measured by its accuracy on the corresponding validation file for that model's language (20% of the training data, as illustrated in Figure 3.5), is logged for each hyperparameter combination. When all combinations are tested, the script automatically identifies the one that yielded the best results for each model. Figure 3.7 illustrates this first stage - hyperparameter search.

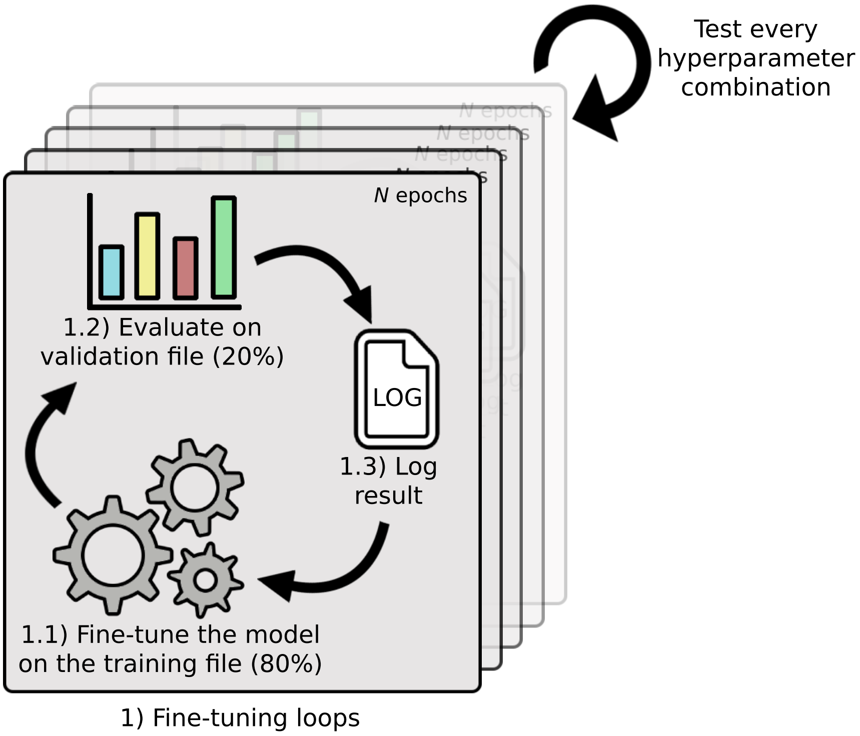

Test every hyperparameter combination
N epochs
1.2) Evaluate on validation file (20%)
LOG
1.3) Log result
1.1) Fine-tune the model on the training file (80%)
1) Fine-tuning loops

Figure 3.7: EL Pipeline Part 1 - Hyperparameter search (for one model).

Once the optimal hyperparameters were identified, the final model fine-tuning was carried out. At this stage, we applied the best hyperparameter combinations to their respective models and initiated the fine-tuning process. Each model was fine-tuned on the full train set, with the resulting predictions extracted from the test set. These predictions were then evaluated using the SympTEMIST official evaluation library, which provided an accurate comparison of the model's performance to the official reported results, across the various languages. Figure 3.8 illustrates the final EL fine-tuning pipeline.

3.4.2 Data Augmentation

For the Entity Linking data augmentation algorithm, we adapted the Named Entity Recognition approach to the structure of EL training files, where each line follows the format (x_1, x_2, *code*) outlined before. The algorithm starts by grouping entity mentions by their codes, and for each code it attempts to augment every entity mention three times (filtering for duplicates in each code), matching the augmentation factor of the NER

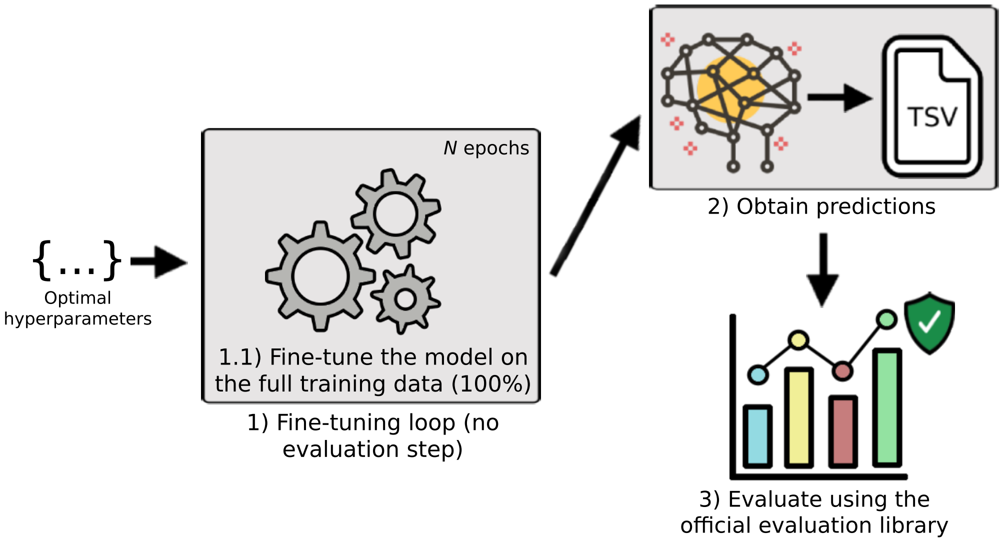

{...}
Optimal hyperparameters
N epochs
1.1) Fine-tune the model on the full training data (100%)
1) Fine-tuning loop (no evaluation step)
TSV
2) Obtain predictions
3) Evaluate using the official evaluation library

Figure 3.8: EL Pipeline Part 2 - Training and evaluating a final model.

algorithm. We again relied on Word2Vec (Mikolov et al., 2013) and FastText (Bojanowski et al., 2017) word embeddings to augment the mentions by generating semantically relevant synonyms.

As with NER, the algorithm replaces a randomly selected word in each entity mention (excluding structure words) with a synonym. However, instead of selecting only the closest synonym, all valid synonyms with a similarity score above the predefined similarity threshold are used, allowing up to five new variations of each mention. This adds even more variety to the dataset, which we could not do in the NER algorithm as we were not only dealing with mentions, but with the sentences in which they appear, and injecting this many new mentions at a time would mean replicating the phrase over and over again, increasing the file sizes exponentially.

To further maintain diversity, once a word in an entity mention has been chosen for augmentation, it won't be chosen again during subsequent augmentations of that mention. Once all words in the mention have already been replaced (or their replacement has been attempted), the augmentation for that mention stops, preventing unnecessary repetition. If no valid synonyms are found for any word in a mention, that mention remains unmodified. Listing 3.5 presents the EL data augmentatin algorithm's pseudocode. Furthermore, tables 3.20 and 3.21 present the impact of different similarity thresholds on the number of mentions and synonyms of the augmented datasets (the number of unique codes remains the same as we did not generate codes). Lastly, table 3.22 also provides examples of some augmented mentions, using both Word2Vec and FastText. Similarly to the example in Table 3.14, we observe that FastText produced variations of the original word, and thus did not alter the original concept, while Word2Vec introduced more diversity, but at the cost of the results being entirely different concepts. This might not be so beneficial for EL as it might be for NER, and we will explore this hypothesis in the next chapter.

```
// group_mentions_by_code(dataset) takes a dataset of format ($x_1$, $x_2$, code) and produces a
//  dictionary where the keys are the codes and the values are the mentions with that code
// find_synonyms(word, embeddings, threshold) returns up to 5 synonyms

Require dataset D, embeddings E, augmentation factor A, similarity thresholds T

grouped_mentions ← group_mentions_by_code(D)
grouped_augmented_mentions ← empty_dictionary()

for each code c ∈ grouped_mentions do
    mentions ← grouped_mentions[c]
    grouped_augmented_mentions[c] ← mentions

    for each mention m ∈ mentions do
        words ← exclude_structure_words(m)
        replaced_words ← ∅

        for i = 1 to A do
            words_to_replace ← words - replaced_words
            if words_to_replace not exists then
                break
            end

            word_to_replace ← choose_random(words_to_replace)
            synonyms ← find_synonyms(word_to_replace, E, T)

            for each synonym s ∈ synonyms do
                e_aug ← replace_with_synonym(m, word_to_replace, synonym)
                if e_aug ∈ grouped_augmented_mentions[c] then
                    continue
                end
                grouped_augmented_mentions[c] ← grouped_augmented_mentions[c] ∪ {e_aug}
            end for

            replaced_words ← replaced_words ∪ word_to_replace
        end for
    end for
end for

return grouped_augmented_mentions
```

Listing 3.5: The pseudocode for the EL data augmentation algorithm.

Table 3.20: The number of unique mentions and total synonyms from the EL training files augmented using Word2Vec, given the similarity thresholds.

Dataset	Language	Similarity Threshold	Unique Mentions	Total Synonyms
SympTEMIST-EL	ES	0.75	5,203	54,655
		0.80	3,227	22,065
		0.85	1,858	6,145
		0.90	1,415	2,781
SympTEMIST-MEL	EN	0.75	1,707	12,968
		0.80	1,001	4,762
		0.85	556	1,005
		0.90	451	557
	FR	0.75	1,381	9,721
		0.80	802	3,809
		0.85	479	1,001
		0.90	388	466
	IT	0.75	1,616	13,132
		0.80	1,037	5,861
		0.85	554	1,410
		0.90	410	517
	PT	0.75	1,386	10,658
		0.80	918	5,557
		0.85	562	1,974
		0.90	343	405

Table 3.21: The number of unique mentions and total synonyms from the EL training files augmented using FastText, given the similarity thresholds.

Dataset	Language	Similarity Threshold	Unique Mentions	Total Synonyms
SympTEMIST-EL	ES	0.75	4,268	38,360
		0.80	2,464	11,209
		0.85	1,763	5,491
		0.90	1,490	3,776
SympTEMIST-MEL	EN	0.75	1,440	8,211
		0.80	842	2,825
		0.85	560	1,032
		0.90	439	538
	FR	0.75	999	5,404
		0.80	608	2,126
		0.85	465	1,047
		0.90	417	779
	IT	0.75	997	5,045
		0.80	621	1,847
		0.85	480	1,053
		0.90	429	760
	PT	0.75	913	4,729
		0.80	547	1,658
		0.85	434	992
		0.90	366	608

Table 3.22: Examples of augmented mentions for EL, from the SympTEMIST-EL training file, using both Word2Vec and FastText with a similarity threshold of 0.75. Blue text indicates word replacement inside the entity mention.

Version	Entity Mention
Original	abdomen sin alteraciones
Word2Vec	tórax sin alteraciones
FastText	abdómen sin alteraciones
Original	aumento del hígado
Word2Vec	aumento del riñón
FastText	aumento del higado

Finally, we proceeded with fine-tuning every LLM from each language, on each one of the four datasets. We employed the optimal hyperparameters obtained before.

4

Results and Discussion

This chapter presents and examines the results of our experiments within the field of multilingual biomedical Named Entity Recognition (NER) and Entity Linking (EL). We divided it into two sections, each one corresponding to one of those two tasks. Those sections are further categorized into transfer learning results, and data augmentation results - first with Word2Vec (Mikolov et al., 2013) and then with FastText (Bojanowski et al., 2017). We try to reason about these results, analyzing how the different approaches may have influenced model performance and what patterns emerge across datasets of the same, and of different languages. At the end of each section, we combine the most successful strategies for each dataset and compile their results. Lastly, it's important to note that all results were obtained using the official evaluation libraries associated with each dataset, ensuring consistency with shared task benchmarks.

4.1 Named Entity Recognition Results

4.1.1 Transfer Learning Results

We begin by presenting the results of the Named Entity Recognition transfer learning experiments. These results are shown in Table 4.1. The way we selected our NER datasets allows us to analyze our approach both within the same language, using different Spanish datasets, and across languages, using various DrugTEMIST variants. We also provide an additional Table 4.2 with the means, medians, and best reported F1-scores for each dataset, to contextualize our performance.

As we can observe, all of our methods have surpassed the mean F1-scores for their respective datasets, which is a positive indicator of the effectiveness of our approach. Furthermore, while we did not surpass the median F1-scores for SympTEMIST-NER (Lima-López, Farré-Maduell, Gasco-Sánchez, et al., 2023) and CANTEMIST-NER (Miranda-Escalada et al., 2020), we did for DisTEMIST and every language variant of DrugTEMIST (Lima-López et al., 2024). Regarding the best reported scores, we did not manage to surpass any, despite being very close on both DrugTEMIST Spanish and English - just 0.11 and 0.53 percentage points (pp) below, respectively. This discrepancy when comparing our

Table 4.1: NER transfer learning results. "1" and "4" refer to the MultiCardioNER experiments, where "1" indicates that the model was trained with only one entity type, and "4" indicates that it was trained with all four types.

Dataset	Precision	Recall	F1-score
SympTEMIST-NER	0.6766	0.6672	0.6719
CANTEMIST-NER	0.7730	0.7977	0.7852
DisTEMIST - 4	0.7743	0.7826	0.7785
DisTEMIST - 1	0.7822	0.7846	0.7834
DrugTEMIST Spanish - 4	0.9024	0.9523	0.9266
DrugTEMIST Spanish - 1	0.9015	0.9267	0.9139
DrugTEMIST English	0.8892	0.9465	0.9170
DrugTEMIST Italian	0.8151	0.8400	0.8274

Table 4.2: The means, medians, and best reported F1-scores for each NER dataset.

Dataset	Mean	Median	Best
SympTEMIST-NER	0.6420	0.7054	0.7477
CANTEMIST-NER	0.7280	0.8010	0.8700
DisTEMIST	0.6956	0.7229	0.8199
DrugTEMIST Spanish	0.7316	0.8502	0.9277
DrugTEMIST English	0.7723	0.8769	0.9223
DrugTEMIST Italian	0.7195	0.8421	0.8842

results to the means, medians and best reported F1-scores across the datasets is intriguing, especially within the Spanish datasets, in which we used the same exact approach. We first presumed that this was likely due to the approaches of the participants of the shared tasks where we obtained a relatively less favourable outcome having a greater level of complexity - they often involve ensembles of transformer-based models (Gallego & Veredas, 2023; García-Pablos et al., 2020), advanced data augmentation techniques (Grazhdanski et al., 2023; Jonker et al., 2023), specialized NER frameworks (Borchert, Llorca, & Schapranow, 2023), multi-task setups with model chaining (García-Pablos et al., 2020), context incorporation at the document level (Borchert, Llorca, & Schapranow, 2023), and the use of custom Spanish biomedical corpora (Han & Tsai, 2020). On the other hand, our initial approach only consists of a simpler transfer learning approach of applying a pre-trained Large Language Model to the dataset.

However, in the DisTEMIST and DrugTEMIST datasets, where other participants also utilized similarly complex strategies, our results were much more competitive, with DrugTEMIST Spanish and English even falling short of the best reported scores by just 0.11pp and 0.53pp respectively, as we pointed out before. This suggests that the complexity of the approach alone does not necessarily explain these performance differences. Instead, we hypothesize that the nature of the datasets themselves, specifically the complexity of the entities they focus on, plays a significant role. This complexity manifests in two closely related ways: entity mention length and detail.

First, by calculating the average length of the entity mentions in both characters and words (presented in Table 4.3), we quickly observe that datasets containing longer mentions tend to produce lower F1-scores. This suggests that as mentions become longer, accurately identifying them becomes more challenging. For instance, fully extracting a complex, multi-word mention like "muesca en la pared posterior del tercio superior del esófago" (symptom) is considerably more difficult than a single-word drug name like "Aspirina" (drug). On the other hand, this increase in length implies that those mentions contain a higher degree of detail, which can also make them much trickier to extract. For example, a symptom-type mention with multiple adjectives and characteristics like "dolor abdominal generalizado a la palpación, más intenso en epigastrio" can easily be extracted as "dolor abdominal generalizado a la palpación" or even as just "dolor abdominal", which are still valid symptoms (with "dolor abdominal" even being a popular symptom on its own in this dataset). On the other hand of the spectrum, drug-type entity mentions often do not have any qualifier words associated with them - as shown by their length in words being close to 1, these mentions typically contain a single word, the drug name.

Table 4.3: Average entity mention length by characters and words from the NER Spanish datasets. "1" and "4" refer to the MultiCardioNER experiments, where "1" indicates that the model was trained with only one entity type, and "4" indicates that it was trained with all four types.

Dataset	Avg Entity Mention Length Chars	Avg Entity Mention Length Words
SympTEMIST-NER	28.80	3.95
CANTEMIST-NER	18.73	2.28
DisTEMIST - 4	25.72	3.20
DisTEMIST - 1	25.72	3.20
DrugTEMIST Spanish - 4	11.50	1.15
DrugTEMIST Spanish - 1	11.50	1.15

These two aspects should increase the likelihood of false positives - mentions that are only partially extracted and thus cannot be counted as correct - and consequently of false negatives - the full mentions that should have been extracted instead. Indeed, we notice in Table 4.4 that the percentages of false positives and negatives follows the same pattern as our F1-score results - SympTEMIST-NER has the highest percentages of false positives and negatives, followed by CANTEMIST-NER and DisTEMIST, and finally DrugTEMIST with significantly fewer false positives and negatives. This is our main hypothesis - that the model performs better in datasets with smaller and less complex mentions.

Table 4.4: Percentage of False Positives and False Negatives from the NER transfer learning approach, regarding the Spanish datasets. "1" and "4" refer to the MultiCardioNER experiments, where "1" indicates that the model was trained with only one entity type, and "4" indicates that it was trained with all four types.

Dataset	% False Positives	% False Negatives
SympTEMIST-NER	32.34%	33.75%
CANTEMIST-NER	22.70%	19.61%
DisTEMIST - 4	22.57%	21.51%
DisTEMIST - 1	21.78%	21.47%
DrugTEMIST Spanish - 4	9.76%	4.52%
DrugTEMIST Spanish - 1	9.85%	7.13%

Moreover, it is important to highlight an anomaly with our CANTEMIST-NER results. As previously pointed out in the Named Entity Recognition Experiments section, the authors of bsc-bio-ehr-es (Carrino et al., 2022) used CANTEMIST-NER to evaluate the model. They reported an F1-score of 0.8340, which is a considerable increase from our reported F1-score of 0.7852. However, this discrepancy can be attributed not to the quality of the approach, but to the different evaluation methods used - we used the official CANTEMIST-NER evaluation library[1] with its provided gold-standard TSV files, whereas the model's authors evaluated their model using the Hugging Face script[2]. Additionally, this script requires their own parsing of the testset into its own CONLL file, further diverging from the official methods, and thus making the comparison between the two results unreliable. Interestingly, we made an experiment and evaluated our approach using the Hugging Face training script and their parsed CONLL test set[3], where we achieved an F1-score of 0.8425, surpassing their result. The only difference in our approach was that we combined the train and dev splits into one pool and then created an 80/20 training and validation split, which seemed to yield a better score. This highlights the critical importance of using the official evaluation library for consistent comparisons across experiments.

Finally, one last remark regarding the Spanish language, particularly MultiCardioNER: as we can see in Table 4.1, and as we reported in our participation paper (Gonçalves & Lamúrias, 2024), training for extracting multiple entity types offered no significant benefit over focusing on specific types. For this reason, we opted not to include those approaches in our data augmentation experiments.

Regarding the non-Spanish subtasks, we can also draw some interesting conclusions by comparing the results across languages for DrugTEMIST. Our Spanish and English

[1] https://github.com/TeMU-BSC/cantemist-evaluation-library

[2] https://github.com/huggingface/transformers/blob/main/examples/pytorch/token-classification/run_ner.py

[3] https://huggingface.co/datasets/PlanTL-GOB-ES/cantemist-ner

models performed similarly, achieving F1-scores of 0.9139 and 0.917, respectively, while the Italian model lagged behind with 0.8274. This suggests a potential discrepancy in the biomedical knowledge of the Italian model compared to the Spanish and English ones, which may have influenced its performance. One potential reason is the difference in the availability of high-quality, annotated biomedical corpora for training these models. Spanish and English have larger language specific corpora in the biomedical domain. In contrast, Italian lacks the same volume of domain-specific resources, evidenced by the fact that BioBIT had to be trained on an Italian machine translation of the original BioBERT dataset as there was no Italian equivalent, which could compromise the quality of the dataset.

4.1.2 Data Augmentation Results

4.1.2.1 Word2Vec Embeddings

We present the results of our Named Entity Recognition experiments using Word2Vec-based data augmentation, in Table 4.5. In addition to the different similarity thresholds, we included a baseline column for easy comparison. We will proceed with evaluating the impact of this augmentation strategy on our transfer learning approach, prioritizing F1-score improvements.

We will start by focusing on the Spanish datasets. As previously stated, we employed the similarity threshold that yielded the best results in the SympTEMIST-NER dataset, to the other Spanish datasets, to maintain consistency across datasets. This similarity threshold was 0.85. As a result, we observed a general increase in F1-scores across nearly all the Spanish datasets, albeit modest (0.77pp-0.9pp), except for DisTEMIST, where the F1-score actually decreased by 0.74pp.

This can likely be attributed to dataset size: smaller datasets usually benefit more from Data Augmentation, due to the introduction of additional variation which helps to prevent overfitting. On the other hand, larger datasets like DisTEMIST may already contain enough variability as it is, and adding further augmented data could introduce noise and actually diminish the system's performance. This might help explain why DisTEMIST, the largest dataset we used by number of unique (9,410) and total (16,675) entity mentions (refer to Tables 3.8, 3.9 and 3.10 for all non-augmented dataset sizes), did not benefit from further diversification - the additional noise introduced through augmentation may indeed have been detrimental. Additionally, employing a higher similarity threshold to the DisTEMIST augmentation process - in this case 0.90 - which would result in a smaller training file, might not be enough to mitigate this issue, following SympTEMIST-NER's performance which actually decreased with a higher threshold. Nevertheless, this warrants further exploration in the future.

On the other hand, smaller datasets like CANTEMIST-NER and DrugTEMIST saw the intended benefit from the augmentation attempt, with slight F1-score improvements. This further suggests that Data Augmentation may indeed be more effective on smaller

Table 4.5: The results from the NER Word2Vec data augmentation experiments. Each dataset's best F1-score is highlighted in bold.

Dataset	Similarity Threshold	Precision	Recall	F1-score
SympTEMIST - NER	-	0.6766	0.6672	0.6719
	0.75	0.6921	0.6505	0.6707
	0.80	0.6891	0.6511	0.6695
	0.85	0.6847	0.6772	**0.6809**
	0.90	0.6779	0.6604	0.6691
CANTEMIST - NER	-	0.7730	0.7977	0.7852
	0.75	-	-	-
	0.80	-	-	-
	0.85	0.7909	0.7974	**0.7941**
	0.90	-	-	-
DisTEMIST	-	0.7822	0.7846	**0.7834**
	0.75	-	-	-
	0.80	-	-	-
	0.85	0.7876	0.7647	0.7760
	0.90	-	-	-
DrugTEMIST Spanish	-	0.9015	0.9267	0.9139
	0.75	-	-	-
	0.80	-	-	-
	0.85	0.9065	0.9371	**0.9216**
	0.90	-	-	-
DrugTEMIST English	-	0.8892	0.9465	0.9170
	0.75	0.9029	0.9402	0.9212
	0.80	0.9013	0.9396	0.9201
	0.85	0.9008	0.9494	**0.9245**
	0.90	0.8975	0.9460	0.9211
DrugTEMIST Italian	-	0.8151	0.8400	0.8274
	0.75	0.8708	0.8839	0.8773
	0.80	0.8767	0.8889	**0.8828**
	0.85	0.8524	0.8728	0.8625
	0.90	0.8585	0.8828	0.8704

datasets. It is also worth considering, in future experiences, whether a smaller similarity threshold be might even more effective for these smaller datasets.

Thus, for the Spanish datasets, we managed to increase the models' performance on all but one dataset, which demonstrates Data Augmentation's potential for enhancing NER results across different subjects, even if all in the general biomedical domain. However, our analysis suggests a training size threshold - between the sizes of DisTEMIST and SympTEMIST-NER - where our NER augmentation strategy is no longer advantageous, and might instead introduce excessive noise. Although further experimentation with the remaining thresholds might be insightful, we did not have the time to conduct these additional tests, leaving them for future work.

Turning our attention to DrugTEMIST's language variants, we observed a performance increase in both the English and Italian datasets, using any of the tested thresholds, which was expected given our aforementioned dataset size hypothesis. Notably, while the English variant's best run experienced an increase in line with the DrugTEMIST Spanish experiment - less than 1pp - the Italian variant exhibited a more significant improvement, with an F1-score increase of 5.54pp - the largest increase across all datasets. While this could suggest that the Italian Word2Vec model might have been able to capture better vector representations for drugs and medicines during training than the other two models, which resulted in better augmentation, we argue that the explanation might be much simpler than that. It is likely that this significant performance increase is the result of the initial lower baseline F1-score for the Italian dataset in the transfer learning portion of the experiment, which left more room for improvement. The Italian model without Data Augmentation performed worse than its Spanish and English counterparts, which resulted in a greater benefit from the added variance, with any of the tested similarity thresholds.

To conclude, we compiled the best Word2Vec augmentation results for each dataset in Table 4.6. For the Spanish datasets, augmentation allowed us to close the gap between our experiments and the median F1-scores for SympTEMIST-NER and CANTEMIST-NER (refer to Table 4.2), despite still falling short of surpassing these scores. Fortunately however, the situation was different for the DrugTEMIST datasets - we came very close to achieving the best reported F1-score for DrugTEMIST Spanish and Italian and even managed to surpass the best reported score for DrugTEMIST English (which used the same model as us, from Yasunaga et al., 2022), further highlighting the effectiveness of Data Augmentation in smaller datasets.

Table 4.6: The best results from the NER Word2Vec data augmentation experiments.

Dataset	Similarity Threshold	Precision	Recall	F1-score
SympTEMIST-NER	0.85	0.6847	0.6772	0.6809
CANTEMIST-NER	0.85	0.7909	0.7974	0.7941
DisTEMIST	-	0.7822	0.7846	0.7834
DrugTEMIST Spanish	0.85	0.9065	0.9371	0.9216
DrugTEMIST English	0.85	0.9008	0.9494	0.9245
DrugTEMIST Italian	0.80	0.8767	0.8889	0.8828

4.1.2.2 FastText Embeddings

Similar to the last section, we present the results of our FastText-based augmentation experiments in Table 4.7 below, which compares the performance of each dataset when different similarity thresholds are considered, and include a "No Augmentation" column as a baseline.

We followed the same approach in the Spanish datasets as with Word2Vec, using the threshold that yielded the best results for SympTEMIST-NER across the other datasets. This time, however, the best performing threshold was lower, 0.75. Nevertheless, we observed the same pattern of an F1-score increase in all Spanish datasets, except for DisTEMIST, where the F1-score decreased. The fact that we see a consistent pattern with this dataset being the only one that did not improve, and in fact saw a drop in performance, further reinforces our previous hypothesis that this dataset already contains sufficient variation, and any additional augmentation seems to introduce noise rather than beneficial diversity.

Additionally, the best-performing threshold for FastText in SympTEMIST-NER, 0.75, being lower than the threshold that worked best for Word2Vec, 0.85, might indicate that the FastText models do not introduce as much noise as the Word2Vec models at lower thresholds. This seems reasonable to us when we consider the differences in vocabulary size and training data of the two architectures. FastText models are trained on a much larger corpus - encompassing Wikipedia[4] and Common Crawl[5], not just Wikipedia like out Word2Vec models - and they have a larger vocabulary - 2,000,000 words, compared to around 500,000 for Word2Vec. This broader base not only provides FastText with a much larger pool of potential and well supported synonyms to choose from when augmenting a word, but also allows it to generate synonyms for a much wider range of words, including rarer and slightly more complex terms. This fact, coupled with its superior ability to handle rare (and even out-of-vocabulary) words much more effectively, means that these synonyms might still be relevant to the given context. Thus, with a lower similarity

[4] https://www.wikipedia.org

[5] http://commoncrawl.org/

Table 4.7: The results from the NER FastText data augmentation experiments. Each dataset's best F1-score is highlighted in bold.

Dataset	Similarity Threshold	Precision	Recall	F1-score
SympTEMIST - NER	-	0.6766	0.6672	0.6719
	0.75	0.6849	0.6724	**0.6786**
	0.80	0.6784	0.6585	0.6683
	0.85	0.6786	0.6659	0.6722
	0.90	0.6649	0.6443	0.6545
CANTEMIST - NER	-	0.7730	0.7977	0.7852
	0.75	0.7961	0.8092	**0.8026**
	0.80	-	-	-
	0.85	-	-	-
	0.90	-	-	-
DisTEMIST	-	0.7822	0.7846	**0.7834**
	0.75	0.7695	0.7742	0.7719
	0.80	-	-	-
	0.85	-	-	-
	0.90	-	-	
DrugTEMIST Spanish	-	0.9015	0.9267	0.9139
	0.75	0.8963	0.9459	**0.9204**
	0.80	-	-	-
	0.85	-	-	-
	0.90	-	-	-
DrugTEMIST English	-	0.8892	0.9465	0.9170
	0.75	0.8942	0.9483	0.9205
	0.80	0.8960	0.9407	0.9178
	0.85	0.8976	0.9471	**0.9217**
	0.90	0.8858	0.9372	0.9108
DrugTEMIST Italian	-	0.8151	0.8400	0.8274
	0.75	0.8620	0.8950	**0.8782**
	0.80	0.8542	0.8856	0.8696
	0.85	0.8616	0.8856	0.8734
	0.90	0.8293	0.8744	0.8513

threshold like 0.75, FastText can leverage its comprehensive knowledge to generate more synonyms for these words, without the majority of them getting filtered out.

On the other hand, Word2Vec's smaller vocabulary and narrower training data likely hinder its ability to capture more complex relationships between words. As a result, a higher similarity threshold - such as 0.85 - is necessary to ensure that the generated synonyms are truly reliable and close in meaning. When working with a lower threshold, Word2Vec might produce more inaccurate synonyms, which introduces noise. For example, the Spanish Word2Vec model lists the symptom "náusea" (nausea) as the fourth synonym for "picazón" (itch), with a similarity score of 0.803, meaning it could eventually be chosen for augmentation with a threshold of 0.80. In contrast, the FastText model does not output this synonym at all.

It is important to note that, because we did not explore the other thresholds apart from 0.75 in the remaining Spanish datasets, these trends, although relatively consistent, are worth investigating further in the context of multilingual biomedical Named Entity Recognition.

We also observed the replication of the DrugTEMIST Italian trend, where we obtained an impressive 5.08pp increase in F1-score. As with the previous experiment, the larger performance gap between the Italian and the other variants, in the transfer learning phase, allowed the augmentation process to have a greater impact in the Italian variant.

To sum up, we present a condensed version of the FastText augmentation results in Table 4.8, to the image of the Word2Vec section, highlighting the best similarity thresholds per dataset. Some results are consistent with those from the Word2Vec section: for SympTEMIST-NER, we managed to close the gap towards the median F1-score, while coming close to the best reported results for DrugTEMIST Spanish and Italian once more. However, despite not surpassing the best reported score for DrugTEMIST English, we actually slightly surpassed the median score for CANTEMIST-NER.

Table 4.8: The best results from the NER FastText data augmentation experiments.

Dataset	Similarity Threshold	Precision	Recall	F1-score
SympTEMIST-NER	0.75	0.6849	0.6724	0.6786
CANTEMIST-NER	0.75	0.7961	0.8092	0.8026
DisTEMIST	-	0.7822	0.7846	0.7834
DrugTEMIST Spanish	0.75	0.8963	0.9459	0.9204
DrugTEMIST English	0.85	0.8976	0.9471	0.9217
DrugTEMIST Italian	0.75	0.8620	0.8950	0.8782

4.1.3 Best Named Entity Recognition Results

To conclude the Named Entity Recognition results, we gathered the best-performing approach for each dataset in Table 4.9, highlighting whether the best result was obtained with Transfer Learning only (no Data Augmentation), Word2Vec augmentation, or FastText augmentation, alongside the corresponding similarity threshold. Interestingly, Word2Vec augmentation outperformed FastText augmentation, despite being trained on much less data. Following this, we provide a broader comparison of the results in Table 4.10. It compares the best reported official F1-scores with our transfer learning scores (without augmentation), and our results from both Word2Vec and FastText data augmentation, across all datasets.

Table 4.9: The best overall results from the NER experiments.

Dataset	Model	Augmentation	Precision	Recall	F1-score
SympTEMIST - NER	PlanTL-GOB-ES/ bsc-bio-ehr-es	Word2Vec - 0.85	0.6847	0.6772	0.6809
CANTEMIST - NER	PlanTL-GOB-ES/ bsc-bio-ehr-es	FastText - 0.80	0.7961	0.8092	0.8026
DisTEMIST	PlanTL-GOB-ES/ bsc-bio-ehr-es	-	0.7822	0.7846	0.7834
DrugTEMIST Spanish	PlanTL-GOB-ES/ bsc-bio-ehr-es	Word2Vec - 0.85	0.9065	0.9371	0.9216
DrugTEMIST English	michiyasunaga/ BioLinkBERT-base	Word2Vec - 0.85	0.9008	0.9494	0.9245
DrugTEMIST Italian	IVN-RIN/bioBIT	Word2Vec - 0.80	0.9008	0.9494	0.9245

Table 4.10: Comparison between the best reported F1-scores, and our F1-scores from the transfer learning, the Word2Vec data augmentation and the FastText data augmentation NER experiments. Values in bold represent an improvement over the best reported score.

Dataset	Best Reported	Transfer Learning	Word2Vec Augmentation	FastText Augmentation
SympTEMIST-NER	**0.7477**	0.6719	0.6809	0.6786
CANTEMIST-NER	**0.8700**	0.7852	0.7941	0.8026
DisTEMIST	**0.8199**	0.7834	0.7834	0.7834
DrugTEMIST Spanish	**0.9277**	0.9139	0.9216	0.9204
DrugTEMIST English	0.9223	0.9170	**0.9245**	0.9217
DrugTEMIST Italian	**0.8842**	0.8274	0.8828	0.8782

4.2 Entity Linking Results

4.2.1 Transfer Learning Results

In this subsection, we present the results of our transfer learning experiments for Entity Linking in Table 4.11, without applying any Data Augmentation, mirroring the beginning of the Named Entity Recognition Results section. The evaluation of our approach was once again conducted using the official evaluation library provided for the task[6], ensuring a faithful assessment of our experiments. We also include another table (Table 4.12) with important benchmarks, namely the mean, median, and best-reported accuracy for each language, to help situate our results and facilitate comparisons.

Table 4.11: EL transfer learning results.

Dataset's Language	Model	Accuracy
ES	cambridgeltl/SapBERT-UMLS-2020AB-all-lang-from-XLMR	**0.5909**
ES	PlanTL-GOB-ES/bsc-bio-ehr-es	0.5643
EN	cambridgeltl/SapBERT-UMLS-2020AB-all-lang-from-XLMR	**0.5938**
EN	emilyalsentzer/Bio_ClinicalBERT	0.5763
EN	michiyasunaga/BioLinkBERT-base	0.5863
EN	microsoft/BiomedNLP-BiomedBERT-base-uncased-abstract-fulltext	0.5881
FR	cambridgeltl/SapBERT-UMLS-2020AB-all-lang-from-XLMR	0.5467
FR	almanach/camembert-bio-base	**0.5495**
FR	quinten-datalab/AliBERT-7GB	0.5382
FR	Dr-BERT/DrBERT-7GB	0.5382
IT	cambridgeltl/SapBERT-UMLS-2020AB-all-lang-from-XLMR	**0.5440**
IT	IVN-RIN/bioBIT	0.5382
IT	IVN-RIN/medBIT	0.5408
IT	IVN-RIN/MedPsyNIT	0.5395
PT	cambridgeltl/SapBERT-UMLS-2020AB-all-lang-from-XLMR	**0.5240**
PT	pucpr/biobertpt-all	0.5062
PT	pucpr/biobertpt-clin	0.5082
PT	neuralmind/bert-base-portuguese-cased	0.5089
PT	PORTULAN/albertina-100m-portuguese-ptpt-encoder	0.5030

[6] https://github.com/nlp4bia-bsc/symptemist_evaluation_library

Table 4.12: The means, medians, and best reported accuracies for each EL dataset.

Dataset's Language	Mean	Median	Best
ES	0.4928	0.5312	0.6070
EN	0.6725	0.7193	0.7250
FR	0.5282	0.5726	0.5733
IT	0.5416	0.5421	0.6703
PT	0.4995	0.5555	0.5575

Regarding the SympTEMIST-EL subtask, focused on Spanish, our approach achieved very competitive results, particularly with the SapBERT model (Liu, Vulić, et al., 2021) where we nearly reached the best-reported accuracy - our result was off by 1.61pp from HPI-DHC's result (Borchert, Llorca, & Schapranow, 2023), the state-of-the-art in this subtask. Their approach also utilized SapBERT, although they did so as part of a larger pipeline - they leveraged the xMEN library (Borchert, Llorca, Roller, et al., 2023), created by the authors themselves for Cross-lingual Medical EL. Their pipeline works in three stages: first, they augment the provided Spanish gazetteer with the training data, as well as with aliases from Spanish UMLS (Bodenreider, 2004). Then, using an ensemble of an untrained SapBERT model and a TF-IDF vectorizer, they generate a list of candidate mentions for each target code. Finally, they fine-tune a BERT-based cross encoder model to re-rank such candidates while maximizing the Accuracy@1, further refining the selection process and improving the overall performance of the ensemble.

Despite not establishing a new state-of-the-art, we managed to outperform the next top-performing teams, BIT.UA (Jonker et al., 2023) and Fusion@SU (Grazhdanski et al., 2023), and place second relative to the task's reported ranking. These teams also leveraged the multilingual SapBERT model[7], but in a more straightforward way - both teams performed cosine similarity search on the SapBERT embeddings from a combined dictionary of the provided gazetteer and training data, much like our approach.

We attribute the success of our approach to two key implementation choices. First is the fact that we augmented our dictionary with the mentions from the training set - both the ones with synonyms (which constitute our training file), and the ones without. As shown in Table 4.13, using a dictionary containing just these mentions alone already provides a remarkably similar result to just using the provided Spanish gazetteer (barely worse by 0.04pp), and combining both sources results in a staggering 11.44pp accuracy boost.

[7] In fact, all but two teams used the multilingual SapBERT model one way or another, and those that didn't use it ended up submitting some of the lowest performing runs, demonstrating why this model is so wide-spread.

Table 4.13: Comparison between different dictionaries. The results were obtained with the cambridgeltl/SapBERT-UMLS-2020AB-all-lang-from-XLMR model, using it's Spanish hyperparameters, described in Table 3.16.

Dictionary	Accuracy
Spanish gazetteer	0.4765
Mentions Present in Training Set	0.4761
Spanish gazetteer + Mentions Present in Training Set	0.5909

However, as we already explained, the two approaches we managed to outperform also incorporated the training data into their dictionaries. Thus, the key factor that allowed us to surpass them was fine-tuning SapBERT on the training set and optimizing its hyperparameters. Table 4.14 extends Table 4.13 by demonstrating the impact of fine-tuning the model on the training data, using different dictionaries.

Table 4.14: Comparison between fine-tuning and not fine-tuning the cambridgeltl/SapBERT-UMLS-2020AB-all-lang-from-XLMR model. The fine-tuning results were obtained using the model's Spanish hyperparameters, described in Table 3.16.

Fine-tuning	Dictionary	Accuracy
No	Spanish gazetteer	0.4547
No	Mentions Present in Training Set	0.4617
No	Spanish gazetteer + Mentions Present in Training Set	0.5674
Yes	Spanish gazetteer	0.4765
Yes	Mentions Present in Training Set	0.4761
Yes	Spanish gazetteer + Mentions Present in Training Set	0.5909

As we can see, fine-tuning the model provides a consistent performance improvement of roughly 1-3pp from just simply applying the model to the testset, regardless of the dictionary. Both of these implementation choices make our approach more robust and complete. Despite of this however, we were not able to outperform the best approach, which may be attributed to its elevated complexity, leading to slightly more refined results. Nevertheless, we believe the simplicity of our methodology to represent a strength against an otherwise elaborate approach, as we achieved comparable results.

For the SympTEMIST-EEL subtask, our performances did not surpass the best-reported accuracies for any of the languages either, which were achieved by the BIT.UA team. They build on their previous methodology, translating all texts into Spanish and applying their aforementioned pipeline from the previous subtask to these newly translated datasets.

This allowed them to leverage the superior dictionary - the combination of the Spanish gazetteer and the training data, as we demonstrated in Table 4.13 - on every language. In contrast, because there are no gazetteers available for any of the language that this subtask comprises, our approach relied solely on the training data to build the corresponding dictionaries.

The translation-based strategy employed by BIT.UA explains their superior performance, especially considering the fact that the SympTEMIST-EEL tracks were primarily designed to explore automatic translation strategies, and the provided non-Spanish datasets were all obtained by translating the Spanish files. Nonetheless, our strategy of using smaller dictionaries still produced relatively strong results, aligning well with this dissertation's theme by demonstrating another viable option for lower-resource languages and scenarios.

Another interesting comparison we can draw from these results is between the monolingual models and the multilingual SapBERT model used across all languages. In general, SapBERT consistently outperformed the monolingual models, except in French, where the CamemBERT-bio model (L. Martin et al., 2020) exceeded SapBERT by 0.28pp. This demonstrates the effectiveness of SapBERT in transferring knowledge from resource-rich languages, specifically English, to resource-poorer languages. Furthermore, the small yet observable margin that the French model CamemBERT-bio has over SapBERT might suggest that this model has the capabilities to outperform it in the French biomedical domain, something that no other monolingual model could. This warrants further exploration.

Finally, it is also insightful to compare the results of the different languages in general: Spanish and English outperformed the other languages by a noticeable margin. Interestingly, English slightly outperformed Spanish by 0.29pp, with a weaker dictionary only comprised of the entity mentions present in the English training set (as the gazetteer was only available in Spanish). This reinforces the advantage of greater resource availability that English has over the other languages, even with a worse dictionary and less training data. The remaining three languages obtained decent results, always surpassing the respective mean accuracy, indicating that while they may lag behind English and Spanish, they still benefited from the methodology and achieved competitive results within their contexts.

4.2.2 Data Augmentation Results

4.2.2.1 Word2Vec Embeddings

In this section, we present the results of applying Word2Vec-based data augmentation to the EL tasks for each language, using different similarity thresholds. These results are revealed in Table 4.15, where we compare performance across various thresholds, including a "no Data Augmentation" column for easier reference, similar to what we did for the NER augmentation experiments. However, unlike those earlier experiments, this time we were able to apply all similarity thresholds to all models across all languages.

This broader application allowed us to better assess the overall impact of our Word2Vec augmentation strategy on the EL datasets.

Table 4.15: The results from the EL Word2Vec data augmentation experiments. "DL" stands for "Dataset's Language". Each model's best accuracy is highlighted in bold.

DL	Model	Similarity Thresholds				
		-	0.75	0.80	0.85	0.90
ES	cambridgeltl/SapBERT-UMLS-2020AB-all-lang-from-XLMR	0.5909	0.5302	0.5527	0.5850	**0.5916**
ES	PlanTL-GOB-ES/bsc-bio-ehr-es	**0.5643**	0.4796	0.5126	0.5471	0.5607
EN	cambridgeltl/SapBERT-UMLS-2020AB-all-lang-from-XLMR	**0.5938**	0.5650	0.5781	0.5919	0.5925
EN	emilyalsentzer/Bio_ClinicalBERT	**0.5763**	0.5625	0.5725	0.5750	0.5725
EN	michiyasunaga/BioLinkBERT-base	0.5863	0.5687	0.5781	**0.5869**	0.5844
EN	microsoft/BiomedNLP-BiomedBERT-base-uncased-abstract-fulltext	0.5881	0.5713	0.5831	**0.5925**	0.5906
FR	cambridgeltl/SapBERT-UMLS-2020AB-all-lang-from-XLMR	0.5467	0.5221	0.5439	0.5481	**0.5495**
FR	almanach/camembert-bio-base	**0.5495**	0.5340	0.5396	0.5467	0.5460
FR	quinten-datalab/AliBERT-7GB	0.5382	0.5130	0.5179	0.5284	**0.5389**
FR	Dr-BERT/DrBERT-7GB	**0.5382**	0.5193	0.5333	0.5368	0.5375
IT	cambridgeltl/SapBERT-UMLS-2020AB-all-lang-from-XLMR	0.5440	0.5188	0.5304	0.5402	**0.5460**
IT	IVN-RIN/bioBIT	0.5382	0.5220	0.5317	0.5337	**0.5402**
IT	IVN-RIN/medBIT	**0.5408**	0.5220	0.5266	0.5337	0.5382
IT	IVN-RIN/MedPsyNIT	**0.5395**	0.5110	0.5220	0.5291	**0.5395**
PT	cambridgeltl/SapBERT-UMLS-2020AB-all-lang-from-XLMR	**0.5240**	0.5003	0.4990	0.5082	0.5181
PT	pucpr/biobertpt-all	0.5062	0.4859	0.4984	0.5036	**0.5102**
PT	pucpr/biobertpt-clin	0.5082	0.4878	0.4944	0.5043	**0.5155**
PT	neuralmind/bert-base-portuguese-cased	0.5089	0.4898	0.4938	0.5030	**0.5115**
PT	PORTULAN/albertina-100m-portuguese-ptpt-encoder	**0.5030**	0.4839	0.4859	0.4964	0.5003

A quick analysis of the results reveals some signs of the same pattern we saw in the NER Word2Vec data augmentation experiments (refer to Table 4.5). Specifically, lower similarity thresholds - 0.75 and 0.80, which naturally allow more variations of the augmented words - consistently hinder performance across all models, without exception. This reinforces our hypothesis that a low similarity threshold does not ensure enough reliability in the synonyms generated by our Word2Vec algorithm and models, as they tend to introduce less accurate variations that negatively affect performance.

The higher similarity thresholds - 0.85 and 0.90 - produced better results, though these were not that remarkable either, especially considering they had much more room to improve than the NER augmentation results to begin with. Despite some visible improvements, particularly with a similarity threshold of 0.90, those gains were modest, averaging around 0.25pp. Furthermore, those improvements only accounted for roughly half of the models, with the rest not benefiting at all from our augmentation approach. The only noteworthy improvements occurred in the Spanish and Italian datasets, where the SapBERT model, despite only achieving 0.07pp and 0.20pp increases in accuracy respectively, it still represented an improvement over our best transfer learning performances for these languages, albeit minimal. As for the rest of the languages, SapBERT saw no improvement except in French, but tied with CamemBERT-bio's transfer learning performance, thus failing to achieve a new highest accuracy. Moreover, no monolingual model experienced enough improvement to surpass SapBERT in its respective language. Table 4.16 presents the condensed version of these results.

Table 4.16: The best results from the EL Word2Vec data augmentation experiments.

Dataset's Language	Model	Similarity Threshold	Accuracy
ES	cambridgeltl/SapBERT-UMLS-2020AB-all-lang-from-XLMR	0.90	0.5916
EN	cambridgeltl/SapBERT-UMLS-2020AB-all-lang-from-XLMR	-	0.5938
FR	cambridgeltl/SapBERT-UMLS-2020AB-all-lang-from-XLMR[8]	0.90	0.5495
IT	cambridgeltl/SapBERT-UMLS-2020AB-all-lang-from-XLMR	0.90	0.5460
PT	cambridgeltl/SapBERT-UMLS-2020AB-all-lang-from-XLMR	-	0.5240

We think this lack of overall improvement has to do with the fact that Entity Linking's nature might make it less susceptible to a data augmentation approach that is not highly sophisticated. This is because, while Named Entity Recognition focuses on identifying entity mentions of a given type, such as drugs or symptoms, which are always relatively general, EL focuses on linking precise medical concepts to their corresponding IDs in a knowledge base. For example, augmenting a mention like "função renal normal" to "função hepática normal" ("normal kidney function" and "normal liver function" respectively), a real augmentation from our Portuguese Word2Vec model, might be beneficial for NER, as it provides an additional example of an entity of type "symptom". However, in EL, these two

[8] The model almanach/camembert-bio-base also achieved an accuracy of 0.5495, but with no augmentation.

mentions would have completely different codes, as they refer to distinct medical concepts (Table 3.14 also provides another example of this phenomenon, where the synonyms generated by Word2Vec fall under the same entity type, but are not the same concept). The specificity required in EL makes it harder to benefit from simpler augmentation approaches, where synonym replacement might introduce more ambiguities and inconsistencies rather than contextually correct synonyms.

4.2.2.2 FastText Embeddings

We now present the results of our FastText-based EL data augmentation approach. These results are displayed in Table 4.17, in a similar table as the previous augmentation sections - it shows the performance of each model across the various datasets, with different similarity thresholds, including the baseline (no augmentation).

Unlike the more consistent trends seen with EL Word2Vec augmentation, the FastText experiments yielded a much more scattered set of outcomes. There was no clear optimal similarity threshold, not even one that worked better than the others enough to be noticeable. Instead, the effectiveness of a given threshold seemed to vary based on which model and which dataset it is applied to. This lack of a clear pattern can be further validated by examining SapBERT's performance across the different languages. For instance, in Spanish and English, SapBERT achieved its best accuracy with a similarity threshold of 0.90. However, in French, the optimal threshold was a tie between 0.90 and 0.80; in Italian, SapBERT performed best with a threshold of 0.75; and in Portuguese, SapBERT's best result was achieved without any augmentation at all.

The monolingual models exhibited a similarly scattered pattern. It is not uncommon to see three or even all four thresholds, each performing best for the different monolingual models in the same language, further emphasizing the varied nature of FastText-based augmentation across models and languages.

Despite these inconsistencies, one similarity with the Word2Vec approach remains: the majority of thresholds actually weaken the performance of the models. However, SapBERT, which was already the top-performing model in the transfer learning experiments, saw at least slight improvements from a different threshold in almost every language. This solidified SapBERT's dominance as the best model, no matter the language, as the condensed results presented in Table 4.18 confirm.

Table 4.17: The results from the EL FastText data augmentation experiments. "DL" stands for "Dataset's Language". Each model's best accuracy is highlighted in bold.

DL	Model	Similarity Thresholds				
		-	0.75	0.80	0.85	0.90
ES	cambridgeltl/SapBERT-UMLS-2020AB-all-lang-from-XLMR	0.5909	0.5811	0.5885	0.5923	**0.5952**
ES	PlanTL-GOB-ES/bsc-bio-ehr-es	**0.5643**	0.5274	0.5407	0.5569	0.5614
EN	cambridgeltl/SapBERT-UMLS-2020AB-all-lang-from-XLMR	0.5938	0.5894	0.5913	0.5887	**0.5969**
EN	emilyalsentzer/Bio_ClinicalBERT	0.5763	**0.5806**	0.5750	0.5713	0.5731
EN	michiyasunaga/BioLinkBERT-base	0.5863	0.5869	0.5863	**0.5900**	0.5869
EN	microsoft/BiomedNLP-Biomed BERT-base-uncased-abstract-fulltext	0.5881	0.5881	**0.5919**	0.5881	0.5887
FR	cambridgeltl/SapBERT-UMLS-2020AB-all-lang-from-XLMR	0.5467	0.5516	**0.5544**	0.5523	**0.5540**
FR	almanach/camembert-bio-base	**0.5495**	0.5481	0.5453	0.5467	0.5460
FR	quinten-datalab/AliBERT-7GB	**0.5382**	0.5333	0.5347	0.5305	0.5319
FR	Dr-BERT/DrBERT-7GB	0.5382	0.5368	0.5347	0.5382	**0.5396**
IT	cambridgeltl/SapBERT-UMLS-2020AB-all-lang-from-XLMR	0.5440	**0.5505**	0.5466	0.5427	0.5473
IT	VN-RIN/bioBIT	0.5382	0.5376	0.5395	0.5421	**0.5427**
IT	VN-RIN/medBIT	0.5408	0.5382	**0.5447**	0.5382	0.5440
IT	VN-RIN/MedPsyNIT	0.5395	0.5389	0.5389	**0.5440**	0.5408
PT	cambridgeltl/SapBERT-UMLS-2020AB-all-lang-from-XLMR	**0.5240**	0.5148	0.5128	0.5181	0.5161
PT	pucpr/biobertpt-all	0.5062	0.5043	0.5023	0.5056	**0.5076**
PT	pucpr/biobertpt-clin	0.5082	0.5089	**0.5102**	0.5062	0.5069
PT	neuralmind/bert-base-portuguese-cased	0.5089	**0.5108**	0.5076	0.5082	0.5102
PT	PORTULAN/albertina-100m-portuguese-ptpt-encoder	0.5030	0.5030	**0.5049**	0.5016	0.4990

Table 4.18: The best results from the EL FastText data augmentation experiments.

Dataset's Language	Model	Similarity Threshold	Accuracy
ES	cambridgeltl/SapBERT-UMLS-2020AB-all-lang-from-XLMR	0.90	0.5952
EN	cambridgeltl/SapBERT-UMLS-2020AB-all-lang-from-XLMR	0.90	0.5969
FR	cambridgeltl/SapBERT-UMLS-2020AB-all-lang-from-XLMR	0.80 and 0.90	0.5544
IT	cambridgeltl/SapBERT-UMLS-2020AB-all-lang-from-XLMR	0.75	0.5505
PT	cambridgeltl/SapBERT-UMLS-2020AB-all-lang-from-XLMR	-	0.5240

4.2.3 Best Entity Linking Results

As with the Named Entity Recognition results section, we compiled the best-performing Entity Linking approach for each language in Table 4.19. This table shows whether the best result was achieved using only Transfer Learning (no Data Augmentation), Word2Vec augmentation, or FastText augmentation, alongside the corresponding similarity threshold for each method.

Table 4.19: The best overall results from the EL experiments.

Dataset's Language	Model	Augmentation	Accuracy
ES	cambridgeltl/SapBERT-UMLS-2020AB-all-lang-from-XLMR	FastText - 0.90	0.5952
EN	cambridgeltl/SapBERT-UMLS-2020AB-all-lang-from-XLMR	FastText - 0.90	0.5969
FR	cambridgeltl/SapBERT-UMLS-2020AB-all-lang-from-XLMR	FastText - 0.80 and 0.90	0.5544
IT	cambridgeltl/SapBERT-UMLS-2020AB-all-lang-from-XLMR	FastText - 0.75	0.5505
PT	cambridgeltl/SapBERT-UMLS-2020AB-all-lang-from-XLMR	-	0.5240

Interestingly, unlike in the NER experiments where Word2Vec consistently outperformed FastText, here FastText emerged as the superior augmentation method. This contrast between the performances of Word2Vec and FastText across NER and EL suggests that, while NER benefits more from general variation, and the augmented mentions can be synonyms inside the same general entity type, EL benefits more from precise synonyms, i.e. synonyms of the specific medical concept. FastText, with its subword-level focus, often generates synonyms by tweaking the morphology of words. For example, in Portuguese, it augmented "massas retroperitoneais" to "massas retroperitoneal," preserving the link to the same medical concept. Word2Vec, on the other hand, does not deal with subword information and may introduce more diverse synonyms that consequently drift too far from the original concept, which risks compromising the entity mention's specific meaning and the link to a specific code.

In essence, FastText's ability to retain the exact concept's meaning made it a slightly better fit for EL tasks. Even so, the limited variation that FastText introduces means it does not create enough diversity inside each medical concept to provide substantial improvements. We believe an ideal augmentation algorithm for EL would likely combine FastText's focus on morphological precision with Word2Vec's potential for broader semantic variation. This is something to be explored in the future.

Finally, we present a broader comparison of all Entity Linking results in Table 4.20. This table contrasts the best official scores reported for each dataset with our best results from Transfer Learning, Word2Vec data augmentation, and FastText data augmentation.

Table 4.20: Comparison between the best reported accuracies, and our accuracies from the transfer learning, the Word2Vec data augmentation and the FastText data augmentation EL experiments.

Dataset's Language	Best Reported	Transfer Learning	Word2Vec Augmentation	FastText Augmentation
ES	**0.6070**	0.5909	0.5916	0.5952
EN	**0.7250**	0.5938	0.5938	0.5969
FR	**0.5733**	0.5495	0.5495	0.5544
IT	**0.6703**	0.5440	0.5460	0.5505
PT	**0.5575**	0.5240	0.5240	0.5240

5

Conclusion

5.1 Overview

This dissertation focused on exploring the tasks Named Entity Recognition (NER) and Entity Linking (EL) for biomedical documents, with an emphasis on lower-resource languages. Our research sought to assess how recent transformer-based models, specifically tailored for the biomedical domain, could be applied to such tasks in order to leverage their pre-existing knowledge (Transfer Learning). Furthermore, we aimed to enhance the performance of these models by leveraging data augmentation techniques, namely synonym replacement using Word2Vec and FastText embeddings.

We first focused on biomedical Named Entity Recognition, a crucial task for extracting relevant entities such as diseases, drugs, and medical terms from free text. We tested our approaches in multiple shared tasks - competitions that provide datasets of different languages and entity types. Our initial transfer learning experiments confirmed that transformer models were indeed a solid choice for this type of task - we were able to achieve results on a par with the best approaches, particularly in DrugTEMIST's English and Spanish variants (Lima-López et al., 2024) - 0.53pp and 1.38pp below the best reported F1-scores, respectively. However, the remaining results were not as remarkable, like in SympTEMIST-NER (Lima-López, Farré-Maduell, Gasco-Sánchez, et al., 2023) and CANTEMIST-NER (Miranda-Escalada et al., 2020) where we didn't crack the median F1-score. We hypothesise that the nature of the entity mentions played a big role in the performance of the models, specifically their length and detail.

Because these results left some room for improvement, we developed a data augmentation technique that takes advantage of Word2Vec (Mikolov et al., 2013) and FastText (Bojanowski et al., 2017) embeddings in order to generate synonyms of the mentions. This technique, particularly when employing Word2Vec embeddings with relatively high similarity thresholds, represented a welcome improvement to our initial transfer learning approach, contributing to some meaningful boosts in performance, particularly the surpassing of the CANTEMIST-NER median F1, or the 5.54pp F1 increase in the DrugTEMIST Italian dataset. Our experiments seem to suggest that Data Augmentation for NER may be more effective on smaller datasets.

We also participated in one of these shared tasks, MultiCardioNER (Gonçalves & Lamúrias, 2024), and although the reported results were unremarkable, we later fixed an issue in the submissions by adding a post-processing step that placed our results among the best for the whole competition.

Next, we shifted our focus to Entity Linking, a task that involves linking recognized mentions to a standardized knowledge base. Using similar methodologies, we applied Transfer Learning and Data Augmentation across SympTEMIST-EL and SympTEMIST-MEL (Lima-López, Farré-Maduell, Gasco-Sánchez, et al., 2023), two EL tasks that provided Spanish, English, French, Italian and Portuguese datasets. Furthermore, we were able to apply both monolingual and multilingual models to these datasets, and compare their results. Regarding the Spanish language, we found that choosing the most complete dictionary, coupled with fine-tuning the model and its hyperparemeters on the task at hand, provided us with praiseworthy results, but not quite enough to achieve a new state-of-the-art as we fell short by 1.18pp. However, our impact was less pronounced on the other languages - we were still able to come close to the best reported accuracy in French (1.89pp), but remained somewhat distant from the best approach in Portuguese (3.35pp), and could not get close in Italian and English (11.98pp and 12.81pp, respectively). Additionally, the fact that the English approach outperformed the Spanish one - even if by just 0.17pp - with SapBERT using a much weaker dictionary, further demonstrates the advantage that the English language has over any other. Finally, in our experiments, the multilingual SapBERT model (Liu, Vulić, et al., 2021) consistently outperformed monolingual models, clearly demonstrating the power of transferring biomedical knowledge from resource-rich languages, mainly English, to resource-poorer ones.

Compared to NER, the impact of Data Augmentation on EL was less pronounced, with only marginal improvements observed - both Word2Vec and FastText-based augmentation yielded modest gains, however FastText emerged as the more effective method in most cases. We hypothesise that, while NER benefits from the broader semantic variation of Word2Vec, EL requires more precise synonym generation, which FastText's subword approach provide.

In conclusion, this dissertation offers a comprehensive analysis of how Transfer Learning, specifically transformer-based models, can be used to great effect on multilingual biomedical Named Entity Recognition and EL. We have also highlighted Data Augmentation's potential for further boosting the performance of these models. Finally, we believe that our research provides valuable insights into the application of transfer learning techniques to the biomedical domain, in a multilingual setting, and hope to contribute to the overall advancement of the field.

5.2 Future Work

We have already been making some remarks and recommendations throughout this dissertation regarding future research that could build on our experiments. This section will summarize those insights, in order to steer future investigation towards addressing gaps in our experiments, or simply opportunities to explore other methodologies.

Firstly, there are a few ways our approach could be improved. For Named Entity Recognition, a clear improvement would be to perform hyperparameter search for each model, as well as applying all similarity thresholds during Data Augmentation, just like we did for Entity Linking. Then, in terms of the EL methodology, securing a larger and more comprehensive dictionary for all languages other than Spanish would likely yield better results, as we saw with the positive impact of the more robust Spanish dictionary. Additionally, we believe that an ideal data augmentation strategy for Entity Linking would involve a combination of FastText's focus on morphological precision with Word2Vec's potential for broader semantic variation. By combining the strengths of the two strategies, we could potentially create a more effective EL data augmentation technique. Lastly, training Word2Vec models on each dataset's specific text, rather than relying solely on Wikimedia data, and fine-tuning the FastText models on that same dataset-specific text (which would act as another form of Transfer Learning), could improve the generated synonyms.

We could also explore a new approach in the future - Multi-Task Learning (MTL). This technique consists in training a model on multiple related tasks simultaneously, allowing it to leverage shared information across tasks. Those tasks could be modelled in several ways. One way is to treat NER as a multi-task problem, where different datasets of the same language (in our case the Spanish datasets) are used for recognizing different entity types in sequence. This would allow the model to first learn to extract entities of easier types, which can then benefit the recognition of more complex ones later (Narayanan et al., 2022). Another way to use MTL would be to model NER and EL tasks together in sequence. For instance, the Vicomtech team showed competitive results when employing this approach on the CANTEMIST-NER dataset, as it allowed the model to use NER as a precursor to EL, improving overall performance (García-Pablos et al., 2020). Another example is Martins et al., 2019, who achieved results competitive with the state-of-the-art in both NER and EL by using MTL to tackle those two tasks together. We initially considered incorporating MTL into our arsenal of approaches, along with Transfer Learning and Data Augmentation, but time constraints prevented us from exploring it.

Lastly, we could extend our exploration to include other datasets, in Spanish (for example the CANTEMIST-EL dataset, which we did not have the opportunity to address) as well as in any other language, in both NER and EL.

Bibliography

Allmuallim, I., Akiba, Y., Yamazaki, T., Yokoo, A., & Kaneda, S. (1994). Two Methods for Learning ALT-J/E Translation Rules from Examples and a Semantic Hierarchy. *COLING 1994 Volume 1: The 15th International Conference on Computational Linguistics*. Retrieved 2024-09-17, from https://aclanthology.org/C94-1006 (cit. on p. 6).

Almeida, T., Jonker, R., Poudel, R., Silva, J. M. F., Jonker, R. A. A., Silva, J. M., & Matos, S. (2023). BIT.UA at MedProcNer: Discovering Medical Procedures in Spanish Using Transformer Models with MCRF and Augmentation. https://doi.org/10.13140/RG.2.2.26577.10080 (cit. on p. 31).

Alsentzer, E., Murphy, J. R., Boag, W., Weng, W.-H., Jin, D., Naumann, T., & McDermott, M. B. A. (2019, June). Publicly Available Clinical BERT Embeddings [arXiv:1904.03323 [cs]]. Retrieved 2024-08-29, from http://arxiv.org/abs/1904.03323 (cit. on pp. 21, 61).

Andrade, V. D., Ruas, P., & Couto, F. M. (2021, September). *Named Entity Recognition and Linking: A Portuguese and Spanish Oncological Parallel Corpus* (preprint). Bioinformatics. https://doi.org/10.1101/2021.09.16.460605 (cit. on p. 2).

Angell, R., Monath, N., Mohan, S., Yadav, N., & McCallum, A. (2021, April). Clustering-based Inference for Biomedical Entity Linking. Retrieved 2024-02-03, from http://arxiv.org/abs/2010.11253 (cit. on pp. 28, 29).

Bahdanau, D., Cho, K., & Bengio, Y. (2016, May). Neural Machine Translation by Jointly Learning to Align and Translate [arXiv:1409.0473 [cs, stat]]. https://doi.org/10.48550/arXiv.1409.0473 (cit. on pp. 7, 8).

Bengio, Y., Ducharme, R., Vincent, P., & Jauvin, C. (2003). A Neural Probabilistic Language Model (cit. on pp. 7, 51).

Berhe, A., Draznieks, G., Martenot, V., Masdeu, V., Davy, L., & Zucker, J.-D. (2023). AliBERT: A Pre-trained Language Model for French Biomedical Text. *The 22nd Workshop on Biomedical Natural Language Processing and BioNLP Shared Tasks*, 223–236. https://doi.org/10.18653/v1/2023.bionlp-1.19 (cit. on p. 61).

Bodenreider, O. (2004). The Unified Medical Language System (UMLS): Integrating biomedical terminology. *Nucleic Acids Research*, *32*(90001), 267D–270. https://doi.org/10.1093/nar/gkh061 (cit. on pp. 1, 26, 58, 83).

Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017, June). Enriching Word Vectors with Subword Information [arXiv:1607.04606 [cs]]. https://doi.org/10.48550/arXiv.1607.04606 (cit. on pp. 4, 52, 67, 71, 92).

Borchert, F., Llorca, I., Roller, R., Arnrich, B., & Schapranow, M.-P. (2023, October). xMEN: A Modular Toolkit for Cross-Lingual Medical Entity Normalization [arXiv:2310.11275 [cs]]. https://doi.org/10.48550/arXiv.2310.11275 (cit. on p. 83).

Borchert, F., Llorca, I., & Schapranow, M.-P. (2023). HPI-DHC @ BC8 SympTEMIST Track: Detection and Normalization of Symptom Mentions with SpanMarker and xMEN. https://doi.org/10.5281/zenodo.10103579 (cit. on pp. 72, 83).

Boudjellal, N., Zhang, H., Khan, A., Ahmad, A., Naseem, R., Shang, J., & Dai, L. (2021). ABioNER: A BERT-Based Model for Arabic Biomedical Named-Entity Recognition (A. Khan, Ed.). *Complexity*, *2021*, 1–6. https://doi.org/10.1155/2021/6633213 (cit. on p. 30).

Brown, P. F., Della Pietra, V. J., deSouza, P. V., Lai, J. C., & Mercer, R. L. (1992). Class-Based *n*-gram Models of Natural Language. *Computational Linguistics*, *18*(4), 467–480. Retrieved 2024-09-17, from https://aclanthology.org/J92-4003 (cit. on p. 7).

Buonocore, T. M., Crema, C., Redolfi, A., Bellazzi, R., & Parimbelli, E. (2023). Localizing in-domain adaptation of transformer-based biomedical language models. *Journal of Biomedical Informatics*, *144*, 104431. https://doi.org/10.1016/j.jbi.2023.104431 (cit. on pp. 44, 61).

Carrino, C. P., Llop, J., Pàmies, M., Gutiérrez-Fandiño, A., Armengol-Estapé, J., Silveira-Ocampo, J., Valencia, A., Gonzalez-Agirre, A., & Villegas, M. (2022). Pretrained Biomedical Language Models for Clinical NLP in Spanish. *Proceedings of the 21st Workshop on Biomedical Language Processing*, 193–199. https://doi.org/10.18653/v1/2022.bionlp-1.19 (cit. on pp. 43–45, 61, 74).

Charniak, E. (1983). Passing Markers: A Theory of Contextual Influence in Language Comprehension* [Publisher: Wiley]. *Cognitive Science*, *7*(3), 171–190. https://doi.org/10.1207/s15516709cog0703_1 (cit. on p. 6).

Chi, Z., Huang, H., Liu, L., Bai, Y., Gao, X., & Mao, X.-L. (2024). Can Pretrained English Language Models Benefit Non-English NLP Systems in Low-Resource Scenarios? *IEEE/ACM Trans. Audio Speech Lang. Process.*, *32*, 1061–1074. https://doi.org/10.1109/TASLP.2023.3267618 (cit. on p. 2).

Chiticariu, L., Krishnamurthy, R., Li, Y., Reiss, F., & Vaithyanathan, S. (2010). Domain Adaptation of Rule-Based Annotators for Named-Entity Recognition Tasks. *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*. https://aclanthology.org/D10-1098 (cit. on p. 24).

Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., Barham, P., Chung, H. W., Sutton, C., Gehrmann, S., Schuh, P., Shi, K., Tsvyashchenko, S., Maynez, J., Rao, A., Barnes, P., Tay, Y., Shazeer, N., Prabhakaran, V., . . . Fiedel, N. (2022, October). PaLM: Scaling Language Modeling with Pathways. Retrieved 2024-02-03, from http://arxiv.org/abs/2204.02311 (cit. on p. 21).

Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L., & Stoyanov, V. (2020, April). Unsupervised Cross-lingual Representation Learning at Scale [arXiv:1911.02116 [cs]]. Retrieved 2024-08-05, from http://arxiv.org/abs/1911.02116 (cit. on pp. 37, 58, 60).

Crema, C., Attardi, G., Sartiano, D., & Redolfi, A. (2022). Natural language processing in clinical neuroscience and psychiatry: A review. *Front Psychiatry*, *13*, 946387. https://doi.org/10.3389/fpsyt.2022.946387 (cit. on p. 1).

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019, May). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Retrieved 2024-02-03, from http://arxiv.org/abs/1810.04805 (cit. on pp. 17, 19–21, 60).

Eltyeb, S., & Salim, N. (2014). Chemical named entities recognition: A review on approaches and applications. *J Cheminform*, *6*(1), 17. https://doi.org/10.1186/1758-2946-6-17 (cit. on p. 24).

Faggioli, G., Ferro, N., Galuščáková, P., & Herrera, A. G. S. d. (Eds.). (2024, September). *Working Notes of CLEF 2024 - Conference and Labs of the Evaluation Forum* (Vol. 3740). CEUR-WS.org. https://ceur-ws.org/Vol-3740/ (cit. on p. 3).

French, E., & McInnes, B. T. (2023). An overview of biomedical entity linking throughout the years. *Journal of Biomedical Informatics*, *137*, 104252. https://doi.org/10.1016/j.jbi.2022.104252 (cit. on pp. 2, 26, 29).

Gallego, F., & Veredas, F. J. (2023). ICB-UMA at BioCreative VIII @ AMIA 2023 Task 2 SYMPTEMIST (Symptom TExt Mining Shared Task) [Place: New Orleans, USA Publisher: Zenodo]. *Proceedings of the BioCreative VIII Challenge and Workshop: Curation and Evaluation in the era of Generative Models*. https://doi.org/10.5281/zenodo.10104058 (cit. on p. 72).

García-Pablos, A., Perez, N., & Cuadros, M. (2020, September). Vicomtech at CANTEMIST 2020 [ISSN: 1613-0073]. In M. Á. G. Cumbreras, J. Gonzalo, E. M. Cámara, R. M. Unanue, P. Rosso, S. J. Zafra, J. A. Ortiz-Zambrano, A. Miranda-Escalada, J. Porta-Zamorano, Y. Guitiérrez, A. Rosá, M. Montes-y-Gómez, & M. García-Vega (Eds.), *Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2020)* (pp. 489–498, Vol. 2664). CEUR. Retrieved 2024-09-18, from https://ceur-ws.org/Vol-2664/#cantemist_paper17 (cit. on pp. 72, 94).

Gonçalves, R., & Lamúrias, A. (2024). Team NOVA LINCS @ BIOASQ12 MultiCardioNER Track: Entity Recognition with Additional Entity Types. *Working Notes of CLEF 2024 - Conference and Labs of the Evaluation Forum*, *3740*, 130–137. https://ceur-ws.org/Vol-3740/paper-11.pdf (cit. on pp. 3, 50, 74, 93).

Grave, E., Bojanowski, P., Gupta, P., Joulin, A., & Mikolov, T. (2018, March). Learning Word Vectors for 157 Languages [arXiv:1802.06893 [cs]]. https://doi.org/10.48550/arXiv.1802.06893 (cit. on pp. 4, 52).

Grazhdanski, G., Vassileva, S., Koychev, I., & Boytcheva, S. (2023). Team Fusion@SU @ BC8 SympTEMIST track: Transformer- based Approach for Symptom Recognition and Linking. https://doi.org/10.5281/zenodo.10103749 (cit. on pp. 72, 83).

Gu, Y., Tinn, R., Cheng, H., Lucas, M., Usuyama, N., Liu, X., Naumann, T., Gao, J., & Poon, H. (2022). Domain-Specific Language Model Pretraining for Biomedical Natural Language Processing. *ACM Trans. Comput. Healthcare*, *3*(1), 1–23. https://doi.org/10.1145/3458754 (cit. on pp. 21, 37, 44, 58, 61).

Guimarães, A., Martins, B., & Magalhães, J. (2024). Lisbon Computational Linguists at SemEval-2024 Task 2: Using A Mistral 7B Model and Data Augmentation [arXiv:2408.03127 [cs]]. *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, 1280–1287. https://doi.org/10.18653/v1/2024.semeval-1.185 (cit. on p. 31).

Guven, Z. A., & Lamurias, A. (2023). Multilingual bi-encoder models for biomedical entity linking. *Expert Systems*, *40*(9), e13388. https://doi.org/10.1111/exsy.13388 (cit. on p. 28).

Han, J.-C., & Tsai, R. T.-H. (2020, September). NCU-IISR: Pre-trained Language Model for CANTEMIST Named Entity Recognition [ISSN: 1613-0073]. In M. Á. G. Cumbreras, J. Gonzalo, E. M. Cámara, R. M. Unanue, P. Rosso, S. J. Zafra, J. A. Ortiz-Zambrano, A. Miranda-Escalada, J. Porta-Zamorano, Y. Guitiérrez, A. Rosá, M. Montes-y-Gómez, & M. García-Vega (Eds.), *Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2020)* (pp. 347–351, Vol. 2664). CEUR. Retrieved 2024-09-18, from https://ceur-ws.org/Vol-2664/#cantemist_paper3 (cit. on p. 72).

Hochreiter, S., & Schmidhuber, J. (1997). Long Short-term Memory. *Neural computation*, *9*, 1735–80. https://doi.org/10.1162/neco.1997.9.8.1735 (cit. on p. 7).

Huang, Z., Xu, W., & Yu, K. (2015, August). Bidirectional LSTM-CRF Models for Sequence Tagging. Retrieved 2024-02-03, from http://arxiv.org/abs/1508.01991 (cit. on p. 24).

Intxaurrondo, A., & Krallinger, M. (2018, November). SPACCC. https://doi.org/10.5281/ZENODO.2560316 (cit. on pp. 38, 45).

Johri, P., Khatri, S. K., Al-Taani, A. T., Sabharwal, M., Suvanov, S., & Kumar, A. (2021). Natural Language Processing: History, Evolution, Application, and Future Work. In A. Abraham, O. Castillo, & D. Virmani (Eds.), *Proceedings of 3rd International Conference on Computing Informatics and Networks* (pp. 365–375, Vol. 167). Springer Singapore. https://doi.org/10.1007/978-981-15-9712-1_31 (cit. on p. 6).

Jonker, R. A. A., Almeida, T., Matos, S., & Almeida, J. (2023). Team BIT.UA @ BC8 SympTEMIST Track: A Two-Step Pipeline for Discovering and Normalizing Clinical Symptoms in Spanish. [Place: New Orleans, USA Publisher: Zenodo]. *Proceedings of the BioCreative VIII Challenge and Workshop: Curation and Evaluation in the era of Generative Models*. https://doi.org/10.5281/zenodo.10103360 (cit. on pp. 72, 83).

Kulshrestha, R. (2021, July). NLP 102: Negative Sampling and GloVe. Retrieved 2024-09-18, from https://towardsdatascience.com/nlp-101-negative-sampling-and-glove-936c88f3bc68 (cit. on p. 53).

Kundeti, S. R., Vijayananda, J., Mujjiga, S., & Kalyan, M. (2016). Clinical named entity recognition: Challenges and opportunities. *2016 IEEE International Conference on Big Data (Big Data)*, 1937–1945. https://doi.org/10.1109/BigData.2016.7840814 (cit. on p. 25).

Labrak, Y., Bazoge, A., Dufour, R., Rouvier, M., Morin, E., Daille, B., & Gourraud, P.-A. (2023, May). DrBERT: A Robust Pre-trained Model in French for Biomedical and Clinical domains [arXiv:2304.00958 [cs]]. Retrieved 2024-08-29, from http://arxiv.org/abs/2304.00958 (cit. on p. 61).

Leal, A., Martins, B., & Couto, F. (2015). ULisboa: Recognition and Normalization of Medical Concepts. *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, 406–411. https://doi.org/10.18653/v1/S15-2070 (cit. on p. 27).

Leaman, R., Islamaj Doğan, R., & Lu, Z. (2013). DNorm: Disease name normalization with pairwise learning to rank. *Bioinformatics*, *29*(22), 2909–2917. https://doi.org/10.1093/bioinformatics/btt474 (cit. on p. 27).

Leaman, R., & Lu, Z. (2016). TaggerOne: Joint named entity recognition and normalization with semi-Markov Models. *Bioinformatics*, *32*(18), 2839–2846. https://doi.org/10.1093/bioinformatics/btw343 (cit. on p. 27).

Lecun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition [Publisher: Institute of Electrical and Electronics Engineers]. *Proceedings of the IEEE*, *86*(11), 2278–2324. https://doi.org/10.1109/5.726791 (cit. on p. 7).

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., & Zettlemoyer, L. (2019, October). BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. Retrieved 2024-02-03, from http://arxiv.org/abs/1910.13461 (cit. on p. 17).

Lima-López, S., Farré-Maduell, E., Gascó, L., Nentidis, A., Krithara, A., Katsimpras, G., Paliouras, G., & Krallinger, M. (2023). Overview of MedProcNER Task on Medical Procedure Detection and Entity Linking at BioASQ 2023. In M. Aliannejadi, G. Faggioli, N. Ferro 0001, & M. Vlachos (Eds.), *Working Notes of the Conference and Labs of the Evaluation Forum (CLEF 2023), Thessaloniki, Greece, September 18th to 21st, 2023* (pp. 1–18, Vol. 3497). CEUR-WS.org. https://ceur-ws.org/Vol-3497/paper-002.pdf (cit. on pp. 31, 45).

Lima-López, S., Farré-Maduell, E., Gasco-Sánchez, L., Rodríguez-Miret, J., & Krallinger, M. (2023). Overview of SympTEMIST at BioCreative VIII: Corpus, guidelines and evaluation of systems for the detection and normalization of symptoms, signs and findings from text. *Proceedings of the BioCreative VIII Challenge and Workshop:*

Curation and Evaluation in the era of Generative Models. https://doi.org/10.5281/ZENODO.10104547 (cit. on pp. 3, 32, 38, 71, 92, 93).

Lima-López, S., Farré-Maduell, E., Rodríguez-Miret, J., Rodríguez-Ortega, M., Lilli, L., Lenkowicz, J., Ceroni, G., Kossoff, J., Shah, A., Nentidis, A., Krithara, A., Katsimpras, G., Paliouras, G., & Krallinger, M. (2024, August). Overview of MultiCardioNER task at BioASQ 2024 on Medical Speciality and Language Adaptation of Clinical NER Systems for Spanish, English and Italian. In G. Faggioli, N. Ferro, P. Galuščáková, & A. García Seco de Herrera (Eds.), *Working Notes of CLEF 2024 - Conference and Labs of the Evaluation Forum* (pp. 8–27, Vol. 3740). CEUR-WS.org. https://ceur-ws.org/Vol-3740/ (cit. on pp. 3, 34, 40, 71, 92).

Liu, F., Shareghi, E., Meng, Z., Basaldella, M., & Collier, N. (2021, April). Self-Alignment Pretraining for Biomedical Entity Representations. Retrieved 2024-02-03, from http://arxiv.org/abs/2010.11784 (cit. on pp. 28–30, 58, 62, 63).

Liu, F., Vulić, I., Korhonen, A., & Collier, N. (2021, May). Learning Domain-Specialised Representations for Cross-Lingual Biomedical Entity Linking. Retrieved 2024-02-03, from http://arxiv.org/abs/2105.14398 (cit. on pp. 2, 28, 30, 58, 61–63, 83, 93).

Louis, A. (2020a, July). A Brief History of Natural Language Processing — Part 1 [Publication Title: Medium]. Retrieved 2024-02-04, from https://medium.com/@antoine.louis/a-brief-history-of-natural-language-processing-part-1-ffbcb937ebce (cit. on p. 6).

Louis, A. (2020b, July). A Brief History of Natural Language Processing — Part 2 [Publication Title: Medium]. Retrieved 2024-02-04, from https://medium.com/@antoine.louis/a-brief-history-of-natural-language-processing-part-2-f5e575e8e37 (cit. on p. 6).

Lourenço, J. M. (2021). *The NOVAthesis LaTeX Template User's Manual*. NOVA University Lisbon. https://github.com/joaomlourenco/novathesis/raw/main/template.pdf (cit. on p. i).

Mahowald, K., Ivanova, A. A., Blank, I. A., Kanwisher, N., Tenenbaum, J. B., & Fedorenko, E. (2023, November). Dissociating language and thought in large language models. Retrieved 2024-02-03, from http://arxiv.org/abs/2301.06627 (cit. on p. 1).

Mansouri, A., Affendey, L. S., & Mamat, A. (2008). Named Entity Recognition Approaches. *8* (cit. on p. 24).

Marimon, M., Gonzalez-Agirre, A., Intxaurrondo, A., Rodríguez, H., Martin, J. L., Villegas, M., & Krallinger, M. (2019, September). Automatic De-identification of Medical Texts in Spanish: The MEDDOCAN Track, Corpus, Guidelines, Methods and Evaluation of Results [ISSN: 1613-0073]. In M. Á. G. Cumbreras, J. Gonzalo, E. M. Cámara, R. M. Unanue, P. Rosso, J. Carrillo-de-Albornoz, S. Montalvo, L. Chiruzzo, S. Collovini, Y. Guitiérrez, S. J. Zafra, M. Krallinger, M. Montes-y-Gómez, R. Ortega-Bueno, & A. Rosá (Eds.), *Proceedings of the Iberian Languages Evaluation*

Forum (IberLEF 2019) (pp. 618–638, Vol. 2421). CEUR-WS.org. Retrieved 2024-08-29, from https://ceur-ws.org/Vol-2421/#MEDDOCAN_overview (cit. on p. 31).

Marrero, M., Urbano, J., Sánchez-Cuadrado, S., Morato, J., & Gómez-Berbís, J. M. (2013). Named Entity Recognition: Fallacies, challenges and opportunities. *Computer Standards & Interfaces*, *35*(5), 482–489. https://doi.org/10.1016/j.csi.2012.09.004 (cit. on p. 25).

Martin, C. E., & Riesbeck, C. K. (1986). Uniform parsing and inferencing for learning. *Proceedings of the Fifth AAAI National Conference on Artificial Intelligence*, 257–261 (cit. on p. 6).

Martin, L., Muller, B., Suárez, P. J. O., Dupont, Y., Romary, L., de la Clergerie, É. V., Seddah, D., & Sagot, B. (2020). CamemBERT: A Tasty French Language Model [arXiv:1911.03894 [cs]]. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 7203–7219. https://doi.org/10.18653/v1/2020.acl-main.645 (cit. on pp. 61, 85).

Martins, P. H., Marinho, Z., & Martins, A. F. T. (2019, July). Joint Learning of Named Entity Recognition and Entity Linking. Retrieved 2024-02-03, from http://arxiv.org/abs/1907.08243 (cit. on pp. 26, 94).

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013, September). Efficient Estimation of Word Representations in Vector Space [arXiv:1301.3781 [cs]]. Retrieved 2024-08-29, from http://arxiv.org/abs/1301.3781 (cit. on pp. 4, 7, 51, 67, 71, 92).

Miranda-Escalada, A., Farré-Maduell, E., & Krallinger, M. (2020, September). Named Entity Recognition, Concept Normalization and Clinical Coding: Overview of the Cantemist Track for Cancer Text Mining in Spanish, Corpus, Guidelines, Methods and Results [ISSN: 1613-0073]. In M. Á. G. Cumbreras, J. Gonzalo, E. M. Cámara, R. M. Unanue, P. Rosso, S. J. Zafra, J. A. Ortiz-Zambrano, A. Miranda-Escalada, J. Porta-Zamorano, Y. Guitiérrez, A. Rosá, M. Montes-y-Gómez, & M. García-Vega (Eds.), *Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2020)* (pp. 303–323, Vol. 2664). CEUR-WS.org. Retrieved 2024-08-29, from https://ceur-ws.org/Vol-2664/#cantemist_overview (cit. on pp. 3, 34, 40, 71, 92).

Miranda-Escalada, A., Farré-Maduell, E., Lima-López, S., Gascó, L., Briva-Iglesias, V., Agüero-Torales, M., & Krallinger, M. (2021, June). The ProfNER shared task on automatic recognition of occupation mentions in social media: Systems, evaluation, guidelines, embeddings and corpora. In A. Magge, A. Klein, A. Miranda-Escalada, M. A. Al-garadi, I. Alimova, Z. Miftahutdinov, E. Farre-Maduell, S. L. Lopez, I. Flores, K. O'Connor, D. Weissenbacher, E. Tutubalina, A. Sarker, J. M. Banda, M. Krallinger, & G. Gonzalez-Hernandez (Eds.), *Proceedings of the Sixth Social Media Mining for Health (#SMM4H) Workshop and Shared Task* (pp. 13–20). Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.smm4h-1.3 (cit. on p. 31).

Miranda-Escalada, A., Gascó, L., Lima-López, S., Farré-Maduell, E., Estrada, D., Nentidis, A., Krithara, A., Katsimpras, G., Paliouras, G., & Krallinger, M. (2022, September).

Overview of DisTEMIST at BioASQ: Automatic detection and normalization of diseases from clinical texts: Results, methods, evaluation and multilingual resources [ISSN: 1613-0073]. In G. Faggioli, N. Ferro, A. Hanbury, & M. Potthast (Eds.), *Proceedings of the Working Notes of CLEF 2022 - Conference and Labs of the Evaluation Forum* (pp. 179–203, Vol. 3180). CEUR-WS.org. Retrieved 2024-08-29, from https://ceur-ws.org/Vol-3180/#paper-11 (cit. on pp. 35, 40).

Morwal, S. (2012). Named Entity Recognition using Hidden Markov Model (HMM). *IJNLC*, *1*(4), 15–23. https://doi.org/10.5121/ijnlc.2012.1402 (cit. on p. 24).

Narayanan, S., Mannam, K., Achan, P., Ramesh, M. V., Rangan, P. V., & Rajan, S. P. (2022). A contextual multi-task neural approach to medication and adverse events identification from clinical text. *Journal of Biomedical Informatics*, *125*, 103960. https://doi.org/10.1016/j.jbi.2021.103960 (cit. on p. 94).

Pappas, D., Malakasiotis, P., & Androutsopoulos, I. (2022, April). Data Augmentation for Biomedical Factoid Question Answering. Retrieved 2024-02-03, from http://arxiv.org/abs/2204.04711 (cit. on p. 31).

Phan, U., & Nguyen, N. (2022). Simple Semantic-based Data Augmentation for Named Entity Recognition in Biomedical Texts. *Proceedings of the 21st Workshop on Biomedical Language Processing*, 123–129. https://doi.org/10.18653/v1/2022.bionlp-1.12 (cit. on p. 31).

Quinlan, J. R. (1986). Induction of decision trees [Publisher: Springer Science and Business Media LLC]. *Machine Learning*, *1*(1), 81–106. https://doi.org/10.1007/bf00116251 (cit. on p. 6).

Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). Improving Language Understanding by Generative Pre-Training. https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf (cit. on pp. 17, 18).

Ramshaw, L. A., & Marcus, M. P. (1995, May). Text Chunking using Transformation-Based Learning [arXiv:cmp-lg/9505040]. Retrieved 2024-08-29, from http://arxiv.org/abs/cmp-lg/9505040 (cit. on p. 42).

Řehůřek, R., & Sojka, P. (2010). Software Framework for Topic Modelling with Large Corpora. *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, 45–50 (cit. on p. 52).

Ruas, P., Lamurias, A., & Couto, F. M. (2020). Linking chemical and disease entities to ontologies by integrating PageRank with extracted relations from literature. *J Cheminform*, *12*(1), 57. https://doi.org/10.1186/s13321-020-00461-4 (cit. on p. 28).

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors [Publisher: Nature Publishing Group]. *Nature*, *323*(6088), 533–536. https://doi.org/10.1038/323533a0 (cit. on p. 7).

Santos, R., Rodrigues, J., Gomes, L., Silva, J., Branco, A., Cardoso, H. L., Osório, T. F., & Leite, B. (2024, March). Fostering the Ecosystem of Open Neural Encoders for

Portuguese with Albertina PT* Family [arXiv:2403.01897 [cs]]. Retrieved 2024-08-29, from http://arxiv.org/abs/2403.01897 (cit. on p. 61).

Schneider, E. T. R., De Souza, J. V. A., Knafou, J., Oliveira, L. E. S. E., Copara, J., Gumiel, Y. B., Oliveira, L. F. A. D., Paraiso, E. C., Teodoro, D., & Barra, C. M. C. M. (2020). BioBERTpt - A Portuguese Neural Language Model for Clinical Named Entity Recognition. *Proceedings of the 3rd Clinical Natural Language Processing Workshop*, 65–72. https://doi.org/10.18653/v1/2020.clinicalnlp-1.7 (cit. on p. 61).

Sekine, S., & Nobata, C. (2004). Definition, dictionaries and tagger for Extended Named Entity Hierarchy. *Proceedings of the Fourth International Conference on Language Resources and Evaluation*. http://www.lrec-conf.org/proceedings/lrec2004/pdf/65.pdf (cit. on p. 24).

Settles, B. (2004). Biomedical named entity recognition using conditional random fields and rich feature sets. *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications - JNLPBA '04*, 104. https://doi.org/10.3115/1567594.1567618 (cit. on p. 24).

Shen, D., Zhang, J., Zhou, G., Su, J., & Tan, C.-L. (2003). Effective adaptation of a Hidden Markov Model-based named entity recognizer for biomedical domain. *Proceedings of the ACL 2003 workshop on Natural language processing in biomedicine -*, *13*, 49–56. https://doi.org/10.3115/1118958.1118965 (cit. on p. 24).

Shliazhko, O., Fenogenova, A., Tikhonova, M., Mikhailov, V., Kozlova, A., & Shavrina, T. (2023, October). mGPT: Few-Shot Learners Go Multilingual. Retrieved 2024-02-03, from http://arxiv.org/abs/2204.07580 (cit. on p. 21).

Singhal, K., Tu, T., Gottweis, J., Sayres, R., Wulczyn, E., Hou, L., Clark, K., Pfohl, S., Cole-Lewis, H., Neal, D., Schaekermann, M., Wang, A., Amin, M., Lachgar, S., Mansfield, P., Prakash, S., Green, B., Dominowska, E., Arcas, B. A. y., . . . Natarajan, V. (2023, May). Towards Expert-Level Medical Question Answering with Large Language Models. Retrieved 2024-02-03, from http://arxiv.org/abs/2305.09617 (cit. on p. 21).

Souza, F., Nogueira, R., & Lotufo, R. (2020a). BERTimbau: Pretrained BERT Models for Brazilian Portuguese [Series Title: Lecture Notes in Computer Science]. In R. Cerri & R. C. Prati (Eds.), *Intelligent Systems* (pp. 403–417, Vol. 12319). Springer International Publishing. https://doi.org/10.1007/978-3-030-61377-8_28 (cit. on p. 61).

Souza, F., Nogueira, R., & Lotufo, R. (2020b, February). Portuguese Named Entity Recognition using BERT-CRF. Retrieved 2024-02-03, from http://arxiv.org/abs/1909.10649 (cit. on p. 30).

Su, P., Peng, Y., & Vijay-Shanker, K. (2021, April). Improving BERT Model Using Contrastive Learning for Biomedical Relation Extraction. Retrieved 2024-02-03, from http://arxiv.org/abs/2104.13913 (cit. on p. 31).

Sung, M., Jeon, H., Lee, J., & Kang, J. (2020, May). Biomedical Entity Representations with Synonym Marginalization. Retrieved 2024-02-03, from http://arxiv.org/abs/2005.00239 (cit. on p. 29).

Sutskever, I., Vinyals, O., & Le, Q. V. (2014, December). Sequence to Sequence Learning with Neural Networks [arXiv:1409.3215 [cs]]. https://doi.org/10.48550/arXiv.1409.3215 (cit. on pp. 7, 8).

Szarvas, G., Farkas, R., & Kocsor, A. (2006). A Multilingual Named Entity Recognition System Using Boosting and C4.5 Decision Tree Learning Algorithms. In D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, M. Naor, O. Nierstrasz, C. Pandu Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M. Y. Vardi, G. Weikum, L. Todorovski, N. Lavrač, & K. P. Jantke (Eds.), *Discovery Science* (pp. 267–278, Vol. 4265). Springer Berlin Heidelberg. https://doi.org/10.1007/11893318_27 (cit. on p. 24).

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2023, August). Attention Is All You Need. Retrieved 2024-02-03, from http://arxiv.org/abs/1706.03762 (cit. on pp. 7, 8, 11, 13, 14, 16).

Wang, X., Han, X., Huang, W., Dong, D., & Scott, M. R. (2019). Multi-Similarity Loss With General Pair Weighting for Deep Metric Learning. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 5017–5025. https://doi.org/10.1109/CVPR.2019.00516 (cit. on p. 58).

Workshop, B., Scao, T. L., Fan, A., Akiki, C., Pavlick, E., Ilić, S., Hesslow, D., Castagné, R., Luccioni, A. S., Yvon, F., Gallé, M., Tow, J., Rush, A. M., Biderman, S., Webson, A., Ammanamanchi, P. S., Wang, T., Sagot, B., Muennighoff, N., . . . Wolf, T. (2023, June). BLOOM: A 176B-Parameter Open-Access Multilingual Language Model. Retrieved 2024-02-03, from http://arxiv.org/abs/2211.05100 (cit. on p. 21).

Wu, L., Petroni, F., Josifoski, M., Riedel, S., & Zettlemoyer, L. (2020, September). Scalable Zero-shot Entity Linking with Dense Entity Retrieval. Retrieved 2024-02-03, from http://arxiv.org/abs/1911.03814 (cit. on p. 28).

Wuebker, J., Peitz, S., Rietig, F., & Ney, H. (2013). Improving Statistical Machine Translation with Word Class Models. *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, 1377–1381. https://aclanthology.org/D13-1138.pdf (cit. on p. 24).

Yasunaga, M., Leskovec, J., & Liang, P. (2022, March). LinkBERT: Pretraining Language Models with Document Links [arXiv:2203.15827 [cs]]. Retrieved 2024-08-29, from http://arxiv.org/abs/2203.15827 (cit. on pp. 44, 61, 77).

Yuan, H., Yuan, Z., Gan, R., Zhang, J., Xie, Y., & Yu, S. (2022, April). BioBART: Pretraining and Evaluation of A Biomedical Generative Language Model. Retrieved 2024-02-03, from http://arxiv.org/abs/2204.03905 (cit. on p. 21).

Zhang, A., Lipton, Z. C., Li, M., & Smola, A. J. (2023). *Dive into Deep Learning*. Cambridge University Press. (Cit. on p. 12).

2024

Entity Recognition and Linking for Biomedical Documents

Rodrigo Gonçalves

NOVA
NOVA SCHOOL OF
SCIENCE & TECHNOLOGY