UNIVERSIDADE DE LISBOA

FACULDADE DE CIÊNCIAS

DEPARTAMENTO DE BIOLOGIA ANIMAL

Applying Deep Learning Extreme Multi-Label Classification to the Biomedical and Multilingual Panoramas

André Daniel Costa das Neves

Mestrado em Bioinformática e Biologia Computacional

Dissertação orientada por:

Professor Doutor Francisco José Moreira Couto

Doutor André Francisco Martins Lamúrias

2020

Acknowledgements

Firstly, I would like to thank both my supervisors, Professor Francisco Couto and André Lamúrias, not only for their guidance and support, but for giving me the opportunity to experience the life of a researcher at LASIGE during this year. A big thanks to my colleagues at LASIGE and to the friends that I made along this journey and with whom I have shared many good moments and conversations, with special thanks to Sofia Conceição, Fábio Neves, Jorge Viana da Cruz, Pedro Santos, Henrique Niza, João Colaço, José Ferrão and Nuno Lopes. To my parents, my aunt Rosarinho, my brother and my sister in law which were a constant presence and were always very supportive, and also to those that have passed away before I could finish this thesis and that have always motivated me to follow this path. Finally, I would like to thank God for giving me the strength to pursue this challenge until the end and for always being a beacon of hope in the most difficult moments.

Resumo

A indexação automática de documentos é um passo fundamental para a organização de dados e para a extração de informação relevante dos mesmos. Esta extração de informação é realizada através de processos de prospecção de texto e de técnicas de processamento de linguagem natural que tornam a linguagem natural perceptível para o computador. Actualmente, muitas das soluções que são aplicadas a estes processos consistem em soluções de aprendizagem automática. No entanto, tem se assistido a um aumento contínuo da aplicação de soluções de aprendizagem profunda em tarefas de prospecção de texto e de processamento de linguagem natural visto que, graças aos desenvolvimentos contínuos ao longo dos últimos anos, estas soluções têm conseguido obter cada vez melhores resultados. Uma dessas técnicas é a classificação multi-rótulo extrema, uma técnica de processamento de linguaguem natural que consiste na indexação de documentos com rótulos pertencentes a um conjunto que pode conter milhares ou mesmo milhões de possíveis rótulos.

Este trabalho apresenta um sistema desenvolvido para as ciências biomédicas e para o domínio multilinguístico, através da adaptação de um algoritmo de classificação multi-rótulo extrema usando aprendizagem profunda. O sistema desenvolvido combina ainda um *software* de reconhecimento de entidades nomeadas com o algoritmo de classificação multi-rótulo extrema de forma a melhorar a atribuição de rótulos aos documentos biomédicos.

Para testar o sistema desenvolvido, participei em três competições internacionais com foco na área das ciências biomédicas, nomeadamente na BioASQ *task* 8a, BioASQ *task* MESINESP e ainda na subtarefa CODING da competição CANTEMIST. O objectivo comum destas três competições consistia na indexação de documentos biomédicos com rótulos pertencentes a um dado vocabulário biomédico. No entanto, enquanto na *task* 8a os dados estavam escritos em Inglês, na *task* MESINESP e na CANTEMIST, os dados biomédicos estavam escritos em Espanhol.

Nas competições da BioASQ, o sistema desenvolvido destacou-se sobretudo nas medidas de precisão, superando a grande maioria dos sistemas e ainda alcançando o 1º lugar por duas semanas consecutivas numa das medidas da BioASQ *task* 8a. Na subtarefa CODING da CANTEMIST, o sistema atingiu uma pontuação de 0.506 na medida mais relevante.

Palavras Chave: Literatura Biomédica, Reconhecimento de Entidade Nomeada, Classificação Multi-Rótulo Extrema, Aprendizagem Profunda, Panorama Multilinguístico.

Abstract

Automatic document indexation is a fundamental step for data organization and information retrieval tasks. Information retrieval can be realized through processes of text mining and natural language processing techniques that make natural language understandable to the computer. Nowadays, most solutions that are applied to these processes use machine learning algorithms. However, thanks to continuous developments through recent years, there has been an increasing usage of deep learning solutions applied to text mining and natural language processing tasks, due to the continuous achievement of better results. One of those techniques is extreme multi-label classification, a natural language processing task consisting in the indexation of documents with labels from a label set that may contain thousands or even millions of possible labels.

This work presents a system developed for the biomedical and multilingual panoramas based on the adaptation of a deep learning extreme multi-label classification algorithm. The developed system also combines a named entity recognition software with the extreme multi-label classification algorithm in order to improve the label classification of the biomedical documents.

To test the developed system, I participated in three international challenges focused on the biomedical sciences, namely in the BioASQ task 8a, BioASQ task MESINESP and in CANTEMIST CODING subtask. The common goal of these three competitions was the indexation of biomedical documents with labels belonging to a specific biomedical vocabulary. However, while the data in task 8a was in English, in task MESINESP and in CANTEMIST the biomedical data was written in Spanish.

In the BioASQ competitions, the system stood out in the precision measures, surpassing most competing systems and achieving the 1st place for two consecutive weeks in one evaluation measure in the BioASQ task 8a. In the CANTEMIST CODING subtask, the system achieved a score of 0.506 in the most relevant measure.

Keywords: Biomedical Literature, Named Entity Recognition, Extreme Multi-Label Classification, Deep Learning, Multilingual Panorama.

Resumo Alargado

A indexação automática de documentos e de dados é fundamental para a organização dos dados e para a extração de informação relevante dos mesmos, o que pode ser essencial para o domínio biomédico, pois permite um acesso rápido e facilitado a esta informação, disponibilizando apenas os dados de interesse ao investigador ou utilizador. Para esta extração de informação relevante, são utilizados processos de prospecção de texto juntamente com técnicas de processamento de linguagem natural que tornam a linguagem natural perceptível para o computador, de forma a este poder processar os dados e retirar significado dos mesmos.

Actualmente, muitas das soluções que são aplicadas à prospecção de texto e ao processamento de linguagem natural consistem em soluções de aprendizagem automática que permitem, não só automatizar o processo de extração de informação do texto, mas também permitem ao computador analisar mais detalhadamente os dados, descobrindo novas características que podem levar a resultados melhorados. No entanto, tem se assistido a um aumento contínuo da aplicação de soluções de aprendizagem profunda em tarefas de prospecção de texto e de processamento de linguagem natural. Como o nome indica, estas soluções apresentam características arquitecturais que levam a uma aprendizagem mais profunda dos dados, como um maior número de camadas internas que permitem uma análise mais detalhada dos dados, descobrindo padrões e reconfigurando a próprios parâmetros internos da rede, levando a uma extração de informação melhorada e mais eficiente. São exemplos destas formas de aprendizagem profunda as redes neuronais convolucionais e recurrentes, cujos desenvolvimentos e aplicações nas mais diversas áreas continuam a gerar mais e melhores resultados.

No entanto, tem-se assistido também ao rápido desenvolvimento de modelos de linguagem de aprendizagem profunda com a representação de palavras por vectores. Este tipo de modelos gera um ou mais vectores para cada palavra com base no contexto em que a mesma se insere, levando a uma compreensão melhorada da informação presente no texto e, consequentemente, melhorando os resultados alcançados pelos sistemas em tarefas de processamento de linguagem natural. Um exemplo de sucesso que influenciou o paradigma das soluções para processamento de linguagem natural foi o BERT *(Bidirectional Encoder Representations from Transformers)*, que levou ao desenvolvimento de modelos similares e específicos para diferentes áreas, como a área biomédica. No entanto, para desenvolver este tipo de modelos de linguagem, é necessária uma grande quantidade de dados de qualidade para o domínio em causa, o que muitas vezes pode não suceder, especialmente em línguas que não a inglesa.

Apesar de tudo, graças aos desenvolvimentos contínuos ao longo dos últimos anos na área da aprendizagem profunda, estas soluções têm conseguido obter cada vez melhores resultados e consequentemente

mais visibilidade por parte da comunidade científica. Uma dessas técnicas é a classificação multi-rótulo extrema. Esta técnica consiste na indexação de documentos com rótulos pertencentes a um conjunto que pode conter milhares ou mesmo milhões de possíveis rótulos, o que representa desafios computacionais para a classificação eficaz dos dados. No entanto, com o desenvolvimento das técnicas de aprendizagem profunda, os algoritmos desenvolvidos para esta técnica de classificação também se tornaram mais eficazes e têm vindo a apresentar cada vez melhores resultados em diversas áreas, apesar de ainda não terem sido aplicados à área das ciências biomédicas nem ao domínio multilínguistico.

Assim, este trabalho apresenta um sistema desenvolvido para as ciências biomédicas e para o domínio multilinguístico, através da adaptação de um algoritmo de classificação multi-rótulo extrema usando aprendizagem profunda chamado X-Transformer. O X-Transformer, inicialmente conhecido por X-BERT, é o primeiro algoritmo que escala o modelo de linguagem BERT à classificação multi-rótulo extrema, o que levou o algoritmo a atingir resultados superiores em conjuntos de dados de referência.

O sistema desenvolvido neste trabalho combina ainda o X-Transformer com o MER *(Minimal Named-Entity Recognizer)*, um *software* de reconhecimento de entidade nomeada usado para identificar no texto termos e relações que estejam presentes num dicionário ou ontologia. Assim, espera-se que esta combinação possa melhorar a atribuição de rótulos por parte do algoritmo de classificação multi-rótulo extrema a documentos biomédicos.

Para testar o sistema desenvolvido, participei em três competições internacionais com foco na área das ciências biomédicas. Duas delas fazem parte da BioASQ, uma conhecida competição anual que se foca no processamento de linguagem natural para a área biomédica. Dentro desta competição, participei na *task* 8a e na primeira edição da *task* MESINESP. O objectivo destas duas competições consistia na indexação de documentos biomédicos com termos pertencentes a um vocabulário organizado desenvolvido para as ciências biomédicos. No entanto, a *task* 8a usava documentos e um vocabulário em inglês, enquanto que a *task* MESINESP era focada no domínio multilínguistico, usando documentos biomédicos em espanhol e um vocabulário organizado também em espanhol.

A terceira competição em que participei foi na primeira edição da competição CANTEMIST, uma competição cujo objectivo era a aplicação de técnicas de prospecção de texto a casos clínicos de oncologia escritos na língua espanhola. Dentro desta competição, existiam várias subtarefas, cada uma dedicada a uma técnica específica de processamento de linguagem natural. Eu participei na subtarefa CODING, em cujo objectivo era a classificação dos casos clínicos usando termos pertencentes a um vocabulário de termos oncológicos.

No final, os resultados obtidos pelo sistema desenvolvido foram promissores. Nas duas competições do BioASQ, o sistema não conseguiu obter valores elevados nas medidas de avaliação mais relevantes, mas destacou-se nas medidas focadas na precisão, onde superou a grande maioria dos sistemas, incluindo os sistemas vencedores, chegando mesmo a alcançar o 1º lugar por duas semanas consecutivas numa das medidas de avaliação do BioASQ *task* 8a. Quanto à subtarefa CODING da CANTEMIST, o sistema atingiu uma pontuação de 0.506 na medida mais relevante. No entanto, os organizadores desta competição não disponibilizaram nem os resultados dos restantes sistemas, nem um valor base, de forma que não foi possível comparar o desempenho do sistema com outros nem com um valor base.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Machine Learning algorithms have been applied in text mining and in other Natural Language Processing (NLP) related tasks, such as Named-Entity Recognition (NER), natural language inference, relation extraction and question answering. These tasks make the natural language understandable to the computer, thus being able to extract meaning, relations and knowledge from the text. The main role of machine learning algorithms in this context is to improve and automate the extraction of information from text sources. However, in recent years traditional machine learning algorithms are constantly being replaced by deep learning methods, such as Convolutional Neural Networks [1, 2], Recurrent Neural Networks [3, 4] or pre-trained language models [5, 6, 7, 8, 9] that are fine-tuned using specific vocabularies and datasets from specific domains like biomedical sciences.

With the creation of specific datasets and corpora in the biomedical area, the effectiveness of deep learning techniques for text mining tasks in this area has increased, achieving better and more accurate results. Consequently, through these advances in deep learning techniques applied to text mining and NLP tasks, and thanks to the increasing amounts and availability of biomedical related datasets, one can access and extract specific information about a certain topic inside the biomedical domain with increased reliability. One example is the automatic label indexation of documents, which can be crucial for biomedical data management and information retrieval tasks. In this task, documents are indexed with labels related with the content of the documents and that belong to a specific vocabulary. In the biomedical panorama, one of the most widely used vocabularies is MeSH (Medical Subject Headings)[1], an ontology developed by the National Library of Medicine[2] that encompasses biomedical and health-related terms. MeSH has been used in several deep learning approaches developed specifically for automatic label indexing of biomedical articles [10, 11].

[1] https://www.nlm.nih.gov/mesh/meshhome.html
[2] https://www.nlm.nih.gov/

However, for a deep learning model to be trained and be able to achieve high scores in different evaluation measures, it requires great amounts of text. This text can come from different sources or domains, like a general domain such as Wikipedia, or from specific domains, like biomedical sciences, that have specific terms and vocabulary related to the domain. The choice of datasets and corpora to use is essential when training the model, since choosing a domain-specific dataset can lead the system to achieve better results than using a general domain dataset [7, 8, 12].

Still, most deep learning models and corpora are designed only for the English language, thus being difficult to apply them to text mining and other NLP tasks in non-English languages. With the reduced number of multilingual datasets, the performance and accuracy of the model drops when using non-English data. This can be even more significative when dealing with specific domains, like biomedical sciences, where structured domain-specific corpora may not exist in non-English languages. Due to this, most developments in text mining and NLP tasks using deep learning are only focused in the English language.

Nonetheless, one promising solution to improve the results of NLP tasks in the multilingual panorama is Extreme Multi-label Classification (XMLC). This technique consists in assigning to text multiple labels from several candidate labels from a dataset that can achieve millions, a challenge for most machine learning approaches, since it requires the model to learn thousands or even millions of labels and features [13]. The choice of labels to use can be from any language and from any domain. Thus, with the right choice of labels, it can be applied to the multilingual panorama or to specific NLP tasks, like semantic indexing or even question answering. XMLC has been receiving an increased focus in recent years, where some deep learning approaches have obtained considerable results in large datasets [14, 15].

Although deep learning models have achieved good results in many scientific community challenges and are constantly being improved, there are not many models that include the use of ontologies and linked data in their core. Most models, like BERT [5] or XLNet [9], have a Transformer architecture [16] which contains a set of pretrained weights that can be used to train a model from large corpora. The model can then be further fine-tuned with external datasets, normally from the specific domain where the model will be applied, along with data from the task where it will be used, but rarely from structured vocabularies such as ontologies. The same applies to XMLC based solutions, which are not usually combined with ontologies or other NLP tools.

1.2 Objectives

XMLC has been receiving an increased focus from the NLP and Text Mining communities, and deep learning XMLC solutions have not yet been applied to the biomedical panorama, where the number of labels in literature can easily reach thousands. Therefore, the main objective of this project was the development of a deep learning XMLC based solution for the biomedical domain, capable of indexing scientific articles with labels related to their content.

To further improve the label indexation, this thesis proposes to combine an XMLC algorithm with

a NER tool capable of identifying ontology terms and their synonyms in the text. The hypothesis was that, by combining a NER tool to identify key terms in the text, the XMLC algorithm would be able to attribute more accurate labels to the texts, since it had a series of additional key terms to describe the text.

Finally, the last objective of this project was focused in applying the developed XMLC solution to the multilingual biomedical panorama. In this objective, there was an increased focus in the Spanish language. First, because it is one of the most spoken languages in the world, surpassing English in the number of native speakers. Second, because of the BioASQ MESINESP (Medical Semantic Indexing in Spanish)[3] competition and the CANTEMIST (CANcer TExt Mining Shared Task – tumor named entity recognition)[4] shared task, two of the three international challenges in which I participated, and which had the objective of automatically indexing Spanish written documents from the biomedical domain with labels related to their content. This way, the developed solution was also pioneer in this specific multilingual domain.

1.3 Methodology

The developed solution was based on adapting a state-of-the-art deep learning XMLC algorithm to the biomedical and multilingual biomedical domains, firstly X-BERT [17] and then its most recent version, X-Transformer [15]. X-BERT and X-Transformer are the first deep learning XMLC solutions to successfully adapt deep learning Transformer models such as BERT [5], to the XMLC panorama. They consist in a three-stage framework that in a first step semantically indexes all labels to clusters using word embeddings. Then, through a deep learning model, indexes each text instance to the most relevant cluster, and finally, ranks the labels from the previously indexed clusters, retrieving the most relevant labels for each text instance.

To test and evaluate the proposed solution, I participated in the 2020 BioASQ competitions, namely in task 8a and task MESINESP, which were focused on the biomedical and multilingual domains. Therefore, a pipeline of Python scripts was developed. The first script preprocessed the data given by the competition organizers, which was composed by thousands of biomedical articles. Then, before creating the files that were to be given as input to the XMLC algorithm, I used a NER tool named MER (Minimal Named-Entity Recognizer) [18, 19] to recognize key terms and synonyms from the DeCS (Health Sciences Descriptors)[5] and MeSH ontologies, which were added to the text. This way, I aimed at improving the label prediction of the XMLC algorithm, since there would be an additional list of relevant keywords for each article.

In the end, a final script processed the results from the XMLC algorithm so that they could have the required format for the competitions. A detailed description of this pipeline, the modifications made in the XMLC algorithm, as well as the results achieved to both competitions can be found in Chapter 3 of this thesis.

[3] https://temu.bsc.es/mesinesp/
[4] https://temu.bsc.es/cantemist/
[5] http://decs.bvsalud.org/I/homepagei.htm

Then, after the BioASQ competitions, a similar pipeline was developed for the CANTEMIST competition, namely the CANTEMIST-CODING task. This pipeline was based on the same pipeline used in the BioASQ competitions, but only using X-Transformer with modifications made due to the competition data, characteristics and requirements. For this competition, I also used data and pre-trained models that were developed for MESINESP and compared the performance of different models trained with different types and amounts of data. A detailed description of this pipeline and results achieved can be found in Chapter 4 of this thesis.

1.4 Contributions

The main goal of this project was the development of an open source semantic indexing system for the biomedical domain based on XMLC [6], that can be applied not only to the English language but also to the multilingual panorama. The system was initially developed from X-BERT [17] and later adapted to X-Transformer [15]. Both are state-of-the-art XMLC deep learning algorithms that were never applied to the biomedical nor to the multilingual domain. The result of my work was a deep learning XMLC based approach to be successfully applied to both domains.

With this system, I was able to participate in three international competitions: BioASQ task 8a, BioASQ task MESINESP, and in the CANTEMIST-CODING task. The participation and results achieved in the BioASQ competitions are described in Chapter 3, while the CANTEMIST-CODING task is tackled in Chapter 4.

Although the system was not able to achieve high F1-scores, it was able to achieve top scores in the precision measures evaluated in the BioASQ tasks, surpassing most competing systems. As to the CANTEMIST-CODING task, the system achieved a score of 0.506, in a scale of 0 to 1, in Mean Average Precision, which was the most relevant measure for this competition.

Following these participations, I have submitted a scientific article [20] for the BioASQ 2020 Workshop which will be held at CLEF 2020 – Conference and Labs of the Evaluation Forum, and another one [21], co-authored by me, for the CANTEMIST 2020 Workshop which will be held at IberLEF 2020 - Iberian Languages Evaluation Forum.

In addition, I have also participated in the Semantic Indexing and Information Retrieval for Health from heterogeneous content types and languages (SIIRH 2020) workshop, held in conjunction with ECIR 2020 (core conference), in which I have submitted and presented a short paper [22] where I propose a possible application of the XMLC system that I have developed, but to multilingual biomedical Question Answering.

The list of submitted and accepted workshop papers is the following:

- André Neves, André Lamúrias, and Francisco M. Couto, ”Extreme Multi-Label Classification applied to the Biomedical and Multilingual Panorama”, in *The Eighth edition of BioASQ: Large-scale*

[6] https://github.com/Cobollero/Multilingual-Biomedical-XMLC-Pipeline

Biomedical Semantic Indexing and Question Answering, 2020 [20].

- Pedro Ruas, André Neves, Vitor M. Andrade, and Francisco M. Couto, ”LasigeBioTM at CANTEMIST: Named entity recognition and normalization of tumour morphology entities and clinical coding of spanish health-related documents”, in *The Iberian Languages Evaluation Forum 2020 (IberLEF 2020)* [21].
- André Neves, André Lamúrias, and Francisco M. Couto, ”Biomedical question answering using extreme multi-label classification and ontologies in the multilingual panorama”, in *The First International Workshop on Semantic Indexing and Information Retrieval for Health from heterogeneous content types and languages (SIIRH)* [22].

1.5 Document Structure

In addition to this first chapter, this document is structured in four chapters that address the following topics:

- **Chapter 2: Related Work**, provides an overview of concepts, approaches and state-of-the-art solutions developed for text mining tasks, with an increased focus on NER and XMLC.
- **Chapter 3: Biomedical Text Indexing - BioASQ Competitions**, presents the system that I developed for the two BioASQ competitions, along with the results achieved, discussion and other post-competition tests.
- **Chapter 4: Clinical Case Indexing - CANTEMIST Competition**, presents the new pipeline that I developed for the CANTEMIST competition, along with the results achieved and corresponding discussion.
- **Chapter 5: Conclusion**, discusses the main conclusions of this work, and presents some directions for future work.

Chapter 2

Related Work

This section contains the related work, key concepts and the state-of-the-art solutions in the fields related to the project. It is organized to firstly introduce in the concepts of text mining and the two related text mining tasks that are tackled in this project: NER and XMLC. Then, Section 2.2 introduces some techniques from NLP and Artificial Intelligence, and Section 2.3 tackles the importance and the developments in the corpora, including multilingual corpora, that is used in many state-of-the-art models. Sectrion 2.4 presents some scientific community competitions that promote advances in biomedical NLP tasks and in other machine learning tasks, along with Section 2.5 that explains the most common evaluation measures in NLP tasks, including the ones used in the competitions that I have participated. Finally, Section 2.6 contains the state-of-the-art solutions developed for NER and XMLC in several domains.

2.1 Text Mining

Text mining is a process of knowledge discovery in which useful and relevant information is extracted from unstructured text documents [23, 24]. It can manage the large number of words and structures in natural language and it is also capable of dealing with the vagueness and ambiguity present in the text [25]. Thanks to these capabilities, text mining presents itself as an interdisciplinary method for information retrieval that encompasses several NLP tasks like NER, document classification and relation extraction. For each task, the information retrieved might have different representations, specific to the task. This way, it is possible to combine the results of different tasks to improve the results of other tasks, like question answering [26, 27].

2.1.1 Named-Entity Recognition

NER is a text mining task with the objective of recognizing mentions of relevant entities in text. NER is essential for many NLP applications such as question answering, machine translation, text summarization and others [28], since it can locate and identifying entities in the text that are related to the intended objective or to the domain of work. An example of general domain categories could be persons, locations,

time or organizations. The entities can be composed by single words or by a sequence of words, that are usually identified by the system through a tokenization process.

There are several approaches for recognizing entities from text. One of these approaches is the rule-based approach, which relies on manually defined rules and does not require annotated data. Inside this approach, there is the dictionary lookup method, which uses a knowledge base or lexicon from a specific domain, such as an ontology, and then matches the text with the terms from the knowledge base [18, 19]. Another type of approach is based on machine learning techniques, such as deep learning, which require a corpus of text data labeled with the entities and their position in the text. The model is then trained using this corpus, so that it can learn to recognize the entities in new texts [29, 30]. There are also hybrid approaches that combine machine learning techniques with rule-based systems [31], which usually consist in a machine learning model that is later fine-tuned with additional linguistic rules to improve its accuracy.

In the biomedical domain, NER is essentially used to recognize biological and chemical entities, such as proteins, genes, species, diseases, chemicals and mutations. However, this task can be challenging since all these entities have unique characteristics that usually require a domain-specific set of features and rules to correctly identify them [32]. In this project, taking into the account the goals of the competitions in which I participated, I decided to use a dictionary approach to identify terms belonging to the ontologies that were used to classify the biomedical documents given in the competitions, along with their synonyms.

2.1.2 Extreme Multi-Label Classification

Multi-label text classification consists in categorizing data by assigning labels to documents, where each document may have different labels assigned to it simultaneously. This is useful to organize and categorize data in several domains including biomedical sciences [33, 34]. In XMLC however, the domains present hundreds of thousands or even millions of possible labels [13], which presents computational challenges to effectively and efficiently classify the data [35].

The labelling and categorization of text data can be crucial for the performance of other NLP tasks, such as question answering, in which the first step is usually to retrieve documents or text relevant to the question. The labeling of data provided by XMLC can be important to improve the model's accuracy, since the labels might act as clues for the model to choose the most relevant documents or fragments of text as answers to questions. However, due to the large number of possible labels, it might be useful to use additional text mining techniques such as NER to improve the performance of XMLC algorithms. Therefore, in this project, a NER approach will be combined with a deep learning XMLC algorithm named X-BERT [17], later renamed X-Transformer [15], which scales pre-trained language models to the XMLC panorama through a framework that combines semantic label indexing, deep neural matching and a ranking component in order to achieve state-of-the-art results in several datasets. The algorithm will be discussed with more detail in subsection 2.6.2.

2.2 Approaches

2.2.1 NLP Techniques

NLP techniques aim at the extraction of meaning from human written text. This section will focus on several approaches that allow the machine to understand natural language and consequently improve the results of NLP tasks.

2.2.1.1 Tokenization

Tokenization consists in breaking sequences of text into smaller fragments called tokens. These tokens can be words, sentences or even whole paragraphs, depending on the splitting criteria chosen. Usually, for most space-limited languages, such as English or Spanish, whitespaces or punctuation marks are often chosen as delimiter, but they might not always be the best splitting operand. For example, in unsegmented languages, such as Chinese or Japanese, word boundaries and formulation are different, thus requiring different splitting operands [36, 37].

Therefore, a set of tokenization rules or models should be defined for a proper tokenization preprocessing of the data. This is useful to handle cases of abbreviations, or to handle text data from specific domains such as the biomedical domain, where there are chemical compounds or genetic material that might be separated by spaces or punctuation marks and where a proper tokenization is crucial for the performance of the system [38, 39].

2.2.1.2 Stemming

Due to the morphological variants of natural language words, stemming is a process that is usually necessary in text mining to reduce the complexity of the text. It consists in converging those variant forms of words into one common representation, its stem or root [36]. For example, ‘are’, ‘am’ and ‘is’ can all be reduced to their common verb ‘be’, which is their stem.

There are several techniques to apply stemming to text [40], some that consist in removing prefixes or suffixes from words such as the popular Porters algorithm [41], in which the suffixes are removed according to a set of rules. Other approaches revolve around statistical analysis, such as the YASS Stemmer which does not need any linguistic information and relies on unsupervised statistical clustering techniques applied to the corpus of text [42].

2.2.2 Machine Learning

Machine Learning is an area of computer science that aims at the development and study of algorithms that allow a computational system to automatically perform tasks by learning patterns and features from the input data. Thanks to these capabilities, machine learning algorithms are versatile and thus can be

applied to diverse areas and applications. Machine learning algorithms can essentially be divided in supervised and unsupervised learning algorithms [43].

Supervised learning algorithms require that a training set containing the examples with the inputs and the desired labeled outputs. This training set will be used to build a mathematical model or classifier, from which it will be able to classify new unlabeled data given as input. Consequently, the predictions made by the classifier will depend on the training set used to train the model. Therefore, if one label is underrepresented in the training set, the model may fail to predict it, while if it is overrepresented, the model may be biased to incorrectly classify the data with that label.

In unsupervised learning algorithms, the training set does not need to be labeled with the expected output, and thus it will learn by discovering patterns and structures in the data. For example, a trained unsupervised learning model can identify portions of data with similar properties and group them together in clusters.

There can also be cases of semi-supervised learning, i.e. a combination of labeled and unlabeled data to train a model. In cases where the amount of labeled data are scarce, semi-supervised learning techniques can be used to improve the results of supervised learning models, since it provides additional training data to train the model [44].

2.2.2.1 Artificial Neural Networks

Artificial Neural Networks are computing systems capable of massive parallel processing that are inspired by Biological Neural Networks, like the human brain.[45, 46] Each cell of the network is an artificial neuron based on its biological homologous, that transmits signals between neurons through synapses. It is through this signal transmission and processing, in conjunction with machine learning methods, that the network can learn. The first model based on this architecture was developed in 1943 by Warren McCulloch and Walter Pitts, that defined through logical and mathematical theorems a neural network capable of solving any computable function [45].

The neurons are organized in layers where each layer processes the signal before sending it to the next layer of neurons. The layers between the input and output layers are called hidden layers and through them, the learning process can be enhanced. For example, by comparing the output of the layer and the desirable output, it is possible to use backpropagation techniques to minimize the difference between the output of the network and the desirable output, which will adjust the internal parameters of the layers to minimize the deviation between the outputs in each iteration, thus getting closer to the desirable output [3, 46, 47].

2.2.2.2 Deep Learning Techniques

Deep learning algorithms are based in artificial neural networks with a greater number of hidden layers. These layers allow the algorithm to discover patterns in large sets of data and then indicate how the algorithm should change its internal parameters layer by layer, i.e. the parameters of the layers are learnt

from the data [48]. These layers can be organized in several types of architectures such as Convolutional Neural Networks (CNN) [1, 2] or Recurrent Neural Networks (RNN) [3, 4].

CNN have fully connected layers, in which all neurons of one layer are connected to all neurons of the next layer and a convolution function is applied in at least one of those layers. As to RNN, they present feedback loops in their structure, allowing the neurons to remember their input in memory, enabling them to be very precise in predicting the next input. They are also able to represent a sequence of events implicitly, exhibiting a dynamic behavior over time, making them advantageous in analyzing sequential data, like a time series, speech or text [49].

2.2.2.3 Language Models

Most deep learning solutions applied to NLP tasks use word embeddings in their core, which are unsupervised learning models trained with large amounts of text to produce vectors that capture the similarity between words, along with their syntactical and semantic information [50]. This way, the model can make accurate guesses about a word's meaning based on past appearances and can also be used to establish analogies between words. Two examples of these word embeddings techniques are Word2vec [51] and GloVe [52]. However, these models may fail in polysemy scenarios since the word will be represented as a single vector even though it might have different meanings.

To tackle these limitations, context dependent language models were developed, which generate different word embeddings for a word based on its position in a sentence. One example of these models is ELMo (Embeddings from Language Models), which generates word vectors that model the syntax and semantics of words, along with their linguistic context, thus being able to tackle polysemy cases. This is possible thanks to a deep learning bidirectional language model that is trained on a large text corpus to generate the word vectors [53]. These vectors can also be added to existing models in order to improve their performance in other NLP tasks.

Another example of this type of models is BERT (Bidirectional Encoder Representations for Transformers) [5], which has greatly changed the solutions applied to most NLP tasks. BERT is used to pretrain language models from large amounts of unlabeled text. It uses a masked language model that randomly masks tokens in the input data and then predicts which is the masked word based on the context of the text. Thanks to this methodology, the model is pretrained with large amounts of data and then it can be further fine-tuned by using specific datasets for a given task. The success of BERT was such that many other models based on BERT started to appear for specific domains, like BioBERT [7] which was specially developed for biomedical NLP tasks using large amounts of biomedical corpora, along with SciBERT [8] which incorporates a large amount of scientific text corpora from Semantic Scholar[1].

In addition, other state-of-the-art models were developed which surpassed the results of the original BERT. Two of those models are RoBERTa (Robustly optimized BERT approach) [6] and XLNet [9]. RoBERTa uses a dynamic masking approach that generates a masking pattern each time a text sequence

[1] https://www.semanticscholar.org/

is fed to the model. This approach has achieved better results than the original masking used by BERT. The RoBERTa model also used larger batch sizes than the original BERT and more than 10 times the amount of data used to train the larger BERT model. On the other hand, XLNet does not mask tokens. Instead, it predicts the tokens from a sequence using combinations of the remaining tokens in that sequence, being the resulting prediction the most likely sequence of tokens. The XLNet models were also trained with larger amounts of data than the BERT models, and with larger batch sizes. In the end, both RoBERTa and XLNet surpassed BERT in benchmark datasets across different NLP tasks.

One of the main sources of pre-trained models, including multilingual models, is the Hugging Face Transformers library[54][2]. Transformers is a Python library that provides several pre-trained deep learning models, such as BERT, RoBERTa, XLNet and others, for several NLP and language generation tasks. The library is enriched by a vast community of collaborators that adapt the existing models or contribute with novel pre-trained models, many of them in non-English languages.

2.3 Vocabularies and Corpora

In order to train deep learning models and obtain good results, it is important to have large quantities of quality text data, such as scientific articles, books or text from webpages available and stored in datasets. One of the most used vocabularies to label scientific articles is MeSH[3]. This is a controlled vocabulary of biomedical words or phrases that are assigned to scientific articles with the purpose of indexing and describing the articles topics [55], thus facilitating user search. The hierarchical structure of MeSH allows a better organization of the related terms and allows users to search on broader, narrower or even in the whole tree of terms. For example, the term 'Endocardium', which corresponds to the innermost layer of the heart, has as direct ancestor the term 'Heart', which has the term 'Cardiovascular System' as its ancestor, and all these terms are part of a bigger group of terms called 'Anatomy Category'.

The MeSH terms are written in English, but a multilingual alternative based on them already exists. The DeCS Terms[4] were developed from MeSH with the goal of providing a unique terminology for searching in Spanish, Portuguese and English. For each MeSH term, there is a corresponding DeCS term. However, DeCS has more than 4000 exclusive terms that are not present in the MeSH hierarchy.

Other than the data and vocabularies available, language models also constitute an important part in the process of developing deep learning models for the biomedical domain, since many of these language models are trained using large amounts of text from this domain. Nevertheless, although there have been numerous breakthroughs in languages models, the great majority is still focused only in the English language. However, BERT as already been applied to the multilingual panorama with the creation of three different models, with the most noticeable being BERT base multilingual cased and BERT base multilingual uncased. The larger model contains 104 languages, composed by the text of the Wikipedia

[2] https://github.com/huggingface/transformers
[3] https://www.nlm.nih.gov/mesh/meshhome.html
[4] http://decs.bvsalud.org/I/homepagei.htm

pages for each of those languages, excluding the user and talk pages. These models achieved state-of-the-art results in the XNLI dataset [56], a corpus designed for language transfer and cross-lingual sentence classification translated in 15 languages by human experts.

Another improvement in the multilingual panorama, namely in the biomedical domain, was a recent work aimed at the development of word embeddings for the Spanish biomedical domain [12]. The authors use a combination of biomedical data written in Spanish retrieved from the SciELO database[5], along with text data from specific topics of Wikipedia, such as pharmacy, medicine or biology. When evaluated and compared with embeddings from a much larger but general domain corpus, their embeddings model achieved better performance values in a NER task, thus showing the importance of using specific domain data to develop language models.

2.4 Biomedical Text Mining Challenges

It is through text mining challenges promoted by the scientific community that many advances in NLP tasks are achieved. In this section, some of the most relevant competitions related to the biomedical panorama or related to this project are described.

2.4.1 BioASQ

BioASQ[6] is a scientific competition that occurs since 2012 and includes a variety of text classification and NLP challenges in the biomedical domain. Several challenges are organized with the objective of developing and advancing the state-of-the-art solutions for large-scale semantic indexing and question answering in the biomedical domain.

The competition is usually split into two distinct tasks. Task A consists in large-scale biomedical indexing, in which the participants must classify new PubMed[7] articles with MeSH terms, before they are manually annotated by human experts. Task B consists in biomedical question answering, in which the participants must respond with different types of answers which can be concepts, articles, snippets, RDF triples, and even exact or ideal answers.

This year, BioASQ introduced the MESINESP task[8], the first non-English task of the challenge co-organized with the Barcelona Supercomputing Center. The objective of this task is similar to Task A, but the scientific articles are written in Spanish, retrieved from the IBECS (Índice Bibliográfico Español en Ciencias de la Salud)[57] and LILACS (Latin American and Caribbean Health Sciences Literature)[9] databases, and must be classified with DeCS terms before being manually annotated by human experts.

[5] https://scielo.org/
[6] http://bioasq.org/
[7] https://pubmed.ncbi.nlm.nih.gov/
[8] https://temu.bsc.es/mesinesp/
[9] https://lilacs.bvsalud.org/

2.4.2 BioCreative

The BioCreative (Critical Assessment of Information Extraction systems in Biology)[10] is a community driven challenge to evaluate text mining and information extraction systems in the biomedical domain. The main motivation of the challenge is not the competitive aspect, but to serve as a way of comparing the methods applied and the community assessment of scientific progress. The challenge started in 2004 and had several editions since then, with different biomedical text mining tasks in each edition such as, gene identification, identification of protein-protein interactions, chemical compound and drug name recognition, text mining chemical-protein interactions, among others.

2.4.3 CANTEMIST

The CANTEMIST (CANcer TExt Mining Shared Task – tumor named entity recognition)[11] challenge first occurred this year, and it is the first shared subtask focused on NER of tumor morphology terms in the Spanish language and in the automatic assignment of eCIE-O-3.1 codes to documents. The eCIE-O-3.1 codes correspond to the Spanish version of the ICD-O (International Classification of Diseases for Oncology) ontology [58]. The eCIE-O-3.1 codes shall be referred just as eCIE-O codes from here on.

The challenge uses a corpus of clinical cases written in Spanish and it is divided into three distinct tasks: NER, NORM and CODING. In the NER task, the objective is to automatically locate and recognise tumor morphology terms. In the NORM task, the objective is returning and normalising all tumor entity mentions along with their corresponding eCIE-O codes. Finally, in the CODING task, the objective is the classification of clinical cases by returning a list of eCIE-O codes ranked by relevance/confidence for each document.

2.4.4 Kaggle

Kaggle[12] is an online data science and machine learning community that allows its users not only to search and publish public datasets and machine learning models, but also to participate in online data science competitions. Many competitions are sponsored by data science companies or research institutes, with goals that can vary from improving existing solutions, through the development of novel solutions for a given problem. The competitions can be from almost any domain, including biomedical sciences, and usually have monetary prizes associated or even recruitment opportunities for the sponsoring companies.

[10] https://biocreative.bioinformatics.udel.edu/
[11] https://temu.bsc.es/cantemist/
[12] https://www.kaggle.com/

2.5 Evaluation Measures

2.5.1 Commonly used measures and general concepts

In order to evaluate the performance of the systems developed for NLP tasks, the trained models are usually applied to gold standard test sets, which are datasets manually annotated by human experts. For example, in the BioASQ task 8a challenge, the gold standard test set corresponded to the weekly list of articles and the corresponding MeSH labels indexed by human experts. However, these labels were not available to the participants during the competition period.

To better understand the evaluation measures that are used in NLP tasks, it is recommended to understand the definitions of True and False Positives, and True and False Negatives. True Positives (TP) correspond to the case when the model prediction is true and in the gold standard it is also true. A False Positive (FP), corresponds to the case when the model prediction is true, but in the gold standard it is false. In contrast, a True Negative (TN) corresponds to a false prediction by the model and in the gold standard it is also false. False Negative (FN) corresponds to a negative prediction by the model, when in the gold standard is true. This relation is expressed in the Confusion Matrix in Figure 2.1.

		Gold Standard Values	
		True	False
Predicted Values	True	TP	FP
	False	FN	TN

Figure 2.1: Confusion matrix showing the relations between the predictions made by the model compared to the values of the gold standard.

Several evaluation measures can then be used to evaluate the performance of the models. One of the most common is accuracy, which takes into account the number of correct predictions over the total number of predictions, as seen in Equation 2.1.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.1}$$

However, accuracy alone is not enough to evaluate the performance of a machine or deep learning model especially when using class-imbalanced datasets where there are classes/labels that can be more frequent than others [59]. In order to get a more detailed and credible evaluation of a model, additional measures are used with the most common being precision, recall and the F1-score or F-measure, which have their corresponding formulas represented in Equation 2.2. Precision evaluates the number of correct predictions over the total number of predictions made, with higher precision values corresponding to

fewer false positives. Recall corresponds to the fraction of correctly identified positive predictions, with high recall values being an indicator of less false negatives in the predictions. The F-score combines both precision and recall to express an overall evaluation of the model performance and corresponds to the harmonic mean of precision and recall.

$$Precision = \frac{TP}{TP + FP} \qquad Recall = \frac{TP}{TP + FN} \qquad F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{2.2}$$

In XMLC tasks, since the algorithms usually deal with large amounts of data with thousands of possible labels, there is usually a focus in evaluating the performance at top K documents. The measure that is mostly used to evaluate XMLC systems is precision at top K (P@K), which considers the number of most relevant labels retrieved by the algorithm among the top K documents. The same logic can also be applied to the evaluation of recall at top K (R@K), which corresponds to the number of correctly identified relevant labels, and F-score at top K (F@K), which takes into account both precision and recall at top K. The formulas used for these measures are similar to the precision, recall and F-score formulas, with the only difference being that the number of TP, FP and FN take into account all the occurrences at the top K documents, as shown in Equation 2.3.

$$P@K = \frac{TP@K}{TP@K + FP@K} \qquad R@K = \frac{TP@K}{TP@K + FN@K} \qquad F@K = 2 \times \frac{P@K \times R@K}{P@K + R@K} \tag{2.3}$$

In the international challenges, there can be measures other than precision, recall and F1-score that are used to evaluate the performance of the systems and that can be specific according to the requested task. In the international competitions that I participated, the measures used to evaluated the performance of the submitted solutions differed between competitions.

2.5.2 BioASQ evaluation measures

In BioASQ task 8a, the evaluation measures were split between flat and hierarchical measures, and in MESINESP the evaluation measures comprised only the flat measures. The flat measures consist in evaluating the assignment of labels to the documents and are based on the number of TP, TN, FP and FN, while the hierarchical measures apply and hierarchical tree-like structure to the labels and the evaluation is based on the label hierarchy through several methods, such as label similarity or label proximity [60]. The flat evaluation measures can be divided into micro and macro averaging. Micro averaging takes into account the classification per document, while macro averaging considers each class [61]. In Equations 2.4 and 2.5 the formulas corresponding to the micro and macro are presented, with N corresponding to the total number of documents.

$$MiP = \frac{\sum_{n=1}^{N} TPn}{\sum_{n=1}^{N} TPn + \sum_{n=1}^{N} FPn} \qquad MiR = \frac{\sum_{n=1}^{N} TPn}{\sum_{n=1}^{N} TPn + \sum_{i=n}^{N} FNn} \qquad MiF = 2 \times \frac{MiP \times MiR}{MiP + MiR} \tag{2.4}$$

$$MaP = \frac{1}{N}\sum_{n=1}^{N}\frac{TPn}{TPn+FPn} \qquad MaR = \frac{1}{N}\sum_{n=1}^{N}\frac{TPn}{TPn+FNn} \qquad MaF = 2 \times \frac{MaP \times MaR}{MaP + MaR} \tag{2.5}$$

As to the hierarchical measures, they take into account the hierarchy of the labels. Usually, the label sets are augmented with the corresponding ancestors, in an attempt to capture the relations between the different classes in the hierarchy [62]. However, the number of ancestor present in the augmented label sets might differ according to the methods used, with some authors choosing to use all ancestors, to others including the descents of the labels as well as the ancestors [63, 64, 65]. However, using all ancestors to hierarchically evaluate the model predictions can be over-penalizing when errors occur in nodes with many ancestors, therefore Kosmopoulos et al. [62] suggested the usage of the lowest common ancestor (LCA) [66] instead. Independent of the number of ancestors used to augment the label sets, the hierarchical precision (HiP), recall (HiR) and F-score (HiF) can be defined as shown in Equation 2.6, with L^{pred} corresponding to the augmented set of predicted labels and L^{true} corresponding to the augmented set of true labels from the gold standard set.

$$HiP = \frac{\left|L^{pred} \bigcap L^{true}\right|}{\left|L^{pred}\right|} \qquad HiR = \frac{\left|L^{pred} \bigcap L^{true}\right|}{\left|L^{true}\right|} \qquad HiF = 2 \times \frac{HiP \times HiR}{HiP + HiR} \tag{2.6}$$

2.5.3 CANTEMIST evaluation measures

In the CANTEMIST challenge, the measures used to evaluate the systems were precision, recall and the F1-score. This measures were used in all three task of the challenge. However, the CODING task had one additional measure, which was Mean Average Precision (MAP).

MAP was the most relevant measure in the CANTEMIST-CODING task and it corresponds to the mean of the average precision scores achieved in each query [67], which in this competition, corresponded to the number of documents to classify. In CANTEMIST, the average precision formula is represented in Equation 2.7. For each document *n*, the average precision is calculated considering the list of labels retrieved by the system for that document ($|L|$), the precision at top *k* labels (P@k), and relevance *rel(k)*, which equals 1 if the *k*-th element of the list is relevant, or 0 if it is not. The value resulting from the cumulative sum is then divided by the total number of relevant labels present in the Gold Standard test set that are associated to the document (*RL*).

$$AveragePrecision(n) = \frac{\sum_{k=1}^{|L|} P@k \times rel(k)}{RL} \tag{2.7}$$

Taking this into account, the formula used in CANTEMIST to evaluate the MAP measure can be described as seen in Equation 2.8, with N corresponding to the total number of queries, or in this case, the total number of documents.

$$MAP = \frac{\sum_{n=1}^{N} AveragePrecision(n)}{N} \tag{2.8}$$

2.6 State-of-the-art Solutions

Most state-of-the-art solutions use deep learning approaches, which have revealed exceptional results in diverse tasks outside of NLP and from different domains of knowledge, like image and speech recognition [68, 69], through predicting the effects of splicing on gene expression [70]. The same is true for NLP tasks, such as NER or even text generation [54], where in recent years there has been a trend to apply deep learning solutions over traditional machine learning algorithms. In fact, in 2011, Collobert et al. [71] presented a solution regarding a unified neural network that learnt mostly from vast amounts of unlabeled data. Their solution outperformed most of the state-of-the-art approaches in that time regarding specific NLP tasks and domains and was shown to be an efficient and versatile solution.

2.6.1 Named-Entity Recognition

2.6.1.1 Rule-based Approaches

Rule-based approaches rely on manually defined rules to identify domain-specific terms in the text. Usually, the terms are stored in lexicons or dictionaries which will be accessed by the algorithm when processing the text sequences. An example of these systems is MER [18, 19], which presents a minimalistic and flexible approach to easily identify terms in the text, only requiring the lexicon and the input text. MER is also capable of entity linking and it can be used with ontology data, thus providing additional information along with each identified term. The computational performance and the results achieved by MER surpassed several other systems, including machine learning approaches and the Bioportal annotator system [72], a popular NER and entity linking tool developed for the biomedical domain.

Another NER system, exclusive for the biomedical domain, was developed by Quimbaya et al. [73] and aimed at improving the identification of terms in electronic health records, which usually present less rigorous text and less quality when compared with scientific medical text. Their system could identify terms in the text even if they were misspelled or presented differences when compared with the original terms from the system's knowledge base. This characteristic allowed the system to achieve improved recall scores, but at the cost of a decreasing its precision.

2.6.1.2 Deep Learning Approaches

There has been a trend to use deep learning techniques in NER, especially bidirectional LSTM (long short-term memory) neural networks, since it allows tokens that are more distant from the target token to influence the classification of each token [28]. Such an example is the LSTM-CRF [74], a deep learning NER based on a bidirectional LSTM RNN and conditional random fields (CRF). The system is language independent and generates word embeddings from the given input text, which are fed into a bidirectional LSTM neural network. The network then returns a representation of the context that is passed to a CRF layer which will return the predictions for each word. This system surpassed other NER models on

two datasets containing independent named entity labels for the English, Spanish, German and Dutch languages.

Another example is Chemlistem [29], which was developed to identify chemical entities in the text for the BioCreative V.5 challenge [30]. The authors developed three different systems using a LSTM neural network: one used a CRF layer, another used character embeddings alongside multiple LSTM layers, and the last one was an ensemble of the other two. The three models achieved top scores in the competition, with the ensemble model achieving the highest score of the three.

With the advent of pre-trained language models such as ELMo [53] and BERT [5], the performance of NER models was improved. An example is Merge and Label [75], which uses a BERT-based model to merge tokens and/or entities into entities and then labels them independently. The model surpassed the other systems in the ACE 2005 corpus [76] and obtained high scores in the OntoNotes v5.0 [77], the largest NER corpus available. Another approach comes from Li et al. [78] which also uses BERT, but tackles NER as a machine reading comprehension task rather than sequence labeling, in which the identification of each entity can be treated as the answer to a factual question. Their approach achieved state-of-the-art results in several benchmark NER datasets in the English language.

2.6.2 Extreme Multi-Label Classification

In the XMLC task, there have been several machine learning solutions developed in the last decade [13]. Among these solutions, one of the most used and competitive algorithm is Parabel [79]. Parabel uses a tree-based approach combined with a bag-of-words method to organize the labels in a given number of clusters. The results achieved by Parabel not only surpass other XMLC tree-based algorithms, but also show considerably faster training and prediction times.

However, only more recently there have been deep learning approaches developed specifically for XMLC. One of the first attempts was the XML-CNN [80], a convolutional neural network that was adapted from a state-of-the-art approach to a multi-class classification task [81]. The architecture of the neural network was adapted with additional layers, one to capture features more precisely from across different regions of text and another to reduce the model size, increasing performance. The loss function was also changed to a binary cross-entropy loss function so it could better rank the labels. The XML-CNN was successful in implementing deep learning to the XMLC task, surpassing most state-of-the-art methods in several datasets.

Another successful approach was AttentionXML [82], which used a bidirectional LSTM RNN with a multi-label attention layer to capture the most relevant parts of the text. However, AttentionXML could not scale well with the largest datasets, so HAXMLNet was created, adding to the same architecture a hierarchical clustering algorithm to divide the labels into smaller clusters, thus being able to work on larger datasets effectively where AttentionXML could not [14].

Lastly, one recent approach is X-BERT [17], the first deep learning approach to apply pre-trained language models, such as BERT, to the XMLC task. X-BERT uses a three-stage framework that firstly,

semantically indexes all the possible labels in clusters using ELMo [53]. Then, using a deep learning Transformer model, it indexes each text instance to the most relevant cluster. Finally, a linear ranker is trained to rank the labels retrieved from the previous cluster indices, modeling the relevance between each text instance and the retrieved labels, and then calculating the corresponding label scores.

X-BERT surpassed other state-of-the-art XMLC methods in four benchmark datasets. However, a newer and improved version of X-BERT has been released recently, renamed X-Transformer [15]. X-Transformer includes additional Transformer models, such as RoBERTa [6] and XLNet [9] and scales them to XMLC. X-Transformer also improves the ranking phase by combining additional sampling strategies to reduce computational complexity and improve the algorithm performance. The results achieved by X-Transformer surpassed the previous results of X-BERT and other XMLC algorithms in the same four benchmark datasets. In addition, the algorithm was also applied to a query recommendation dataset from Amazon that suggests relevant products to the users according to previously viewed products, product information and how many times they were viewed by the user. In this dataset, X-Transformer showed improvements of more than 10% over Parabel [79], one of the most commonly used and competitive XMLC algorithm.

Although X-BERT and X-Transformer were applied to benchmark datasets, neither of them were applied to the biomedical panorama nor have been combined with ontologies. Therefore, this thesis project explores this opportunity to apply a deep learning XMLC algorithm to the biomedical sciences, aiming at improving the state-of-the-art results.

Chapter 3

Biomedical Text Indexing – BioASQ Competitions

Document indexing is a crucial step for many NLP tasks, such as multi-label classification or question answering, since it can improve the performance of the trained models. As a starting point for this project, I created a text classification deep learning model for biomedical text by adapting the existing XMLC algorithm X-BERT and later its most recent version, X-Transformer. The effectiveness of this method was first tested in the BioASQ task 8a, which consisted in indexing journal abstracts written in English using the MeSH terms, and later on the BioASQ MESINESP task, which had similar goals, but the abstracts were written in Spanish and they were indexed using the DeCS terms. Therefore, this adaptation was also focused on the multilingual panorama, so that the project could contribute to a possible improvement on biomedical multilingual text classification. For both tasks, a pipeline of scripts was developed to process the data given in each task and making it readable for the XMLC algorithm.

In this chapter, I will start by stating the modifications made to the XMLC algorithms so that they could be applied to both competitions. Then, in Section 3.2 I will explain in detail the developed pipeline used in the competitions as well as the developed models in Section 3.5. The model evaluation and the results achieved in the two competitions are present in Sections 3.6 and 3.7. Finally, the results are discussed in Section 3.8, along with some post-competition experiments in Section 3.9.

3.1 X-BERT and X-Transformer modifications

For both tasks, some modifications in the algorithm code were required. The first one was made in the vectorization of the labels of the training, test and validation sets. I have chosen to use all possible labels, including the labels that were not present in neither train, test nor validation sets. Otherwise the algorithm would fail to work correctly if the number of labels between sets did not match.

Another modification was the inclusion of SciBERT [8] in the choices of models to train X-BERT, so that I could use this model in task 8a. This inclusion was not made in X-Transformer, since it was only

used for the MESINESP task which was focused on the Spanish language and SciBERT was exclusive for the English language. X-Transformer required additional changes since, at the time of the MESINESP competition, the algorithm was still being developed by the X-BERT team and it could only process the four datasets in which X-BERT was tested [17]. To surpass this limitation, I have modified some preprocessing scripts from the original X-BERT code and implemented them in X-Transformer, so it could process custom datasets.

Finally, in addition to these modifications, I changed the file encodings in both X-BERT and X-Transformer to utf-8, so that the algorithms could process input data containing diacritical marks, such as accents, that are common in the Spanish language.

3.2 Developed Pipeline

I developed a common pipeline for both competitions, with some changes according to the algorithm used or with the competition, as it can be seen in Figures 3.1 and 3.2. The first step, was to retrieve the data given by the competition organizers. The articles were split into training, test and validation sets using a 60%-20%-20% proportion to be used by X-BERT. The validation set was be used for hyperparameter tuning such as the number of training epochs, since X-BERT performs an evaluation during training and then saves the hyperparameters in a checkpoint if the evaluation results surpassed the results of the previous checkpoint. The values differ when using X-Transformer, because it only requires training and test sets. The test set will also be used by X-Transformer to save the model checkpoints during training evaluation. Therefore, I have decided to use a proportion of 70%-30% for X-Transformer.

The next step was the creation of a vocabulary file, which is required by both X-BERT and X-Transformer and that contains the labels used to classify the data and a corresponding numerical identifier. For that, each line of the vocabulary file has a DeCS or MeSH code, according to the competition, and its corresponding internal identifier, which corresponds to a number from 0 to N, where N is the total number of DeCS or MeSH Terms minus 1, since it starts in 0. This internal numerical identifier is the characteristic that allows X-BERT and X-Transformer to work with any type of label and with any kind of language, since the algorithms will use these numeric identifiers to classify the text. In addition, a label mapping file was created for each competition. For task MESINESP, each line of the label mapping file contains the correspondence between the DeCS term, its code and its numeric identifier in the vocabulary file, while for task 8a, the file has the same structure but using the MeSH terms instead. For example, the term ‘Temefós’, which has the corresponding DeCS code ‘2’, is the first element in the vocabulary file, thus its numeric identifier will be ‘0’. This label mapping file will later be used to map the predictions from their numeric identifiers to the corresponding DeCS or MeSH codes required for the competitions.

After converting the codes to their corresponding numeric identifiers, MER [18, 19] was executed on each article’s abstracts to identify biomedical terms listed in lexicons created by me. For task 8a, the lexicon I created was composed by the MeSH terms and their synonyms. As for task MESINESP, the lexicon created contained the DeCS terms and their synonyms, but I also decided to include their

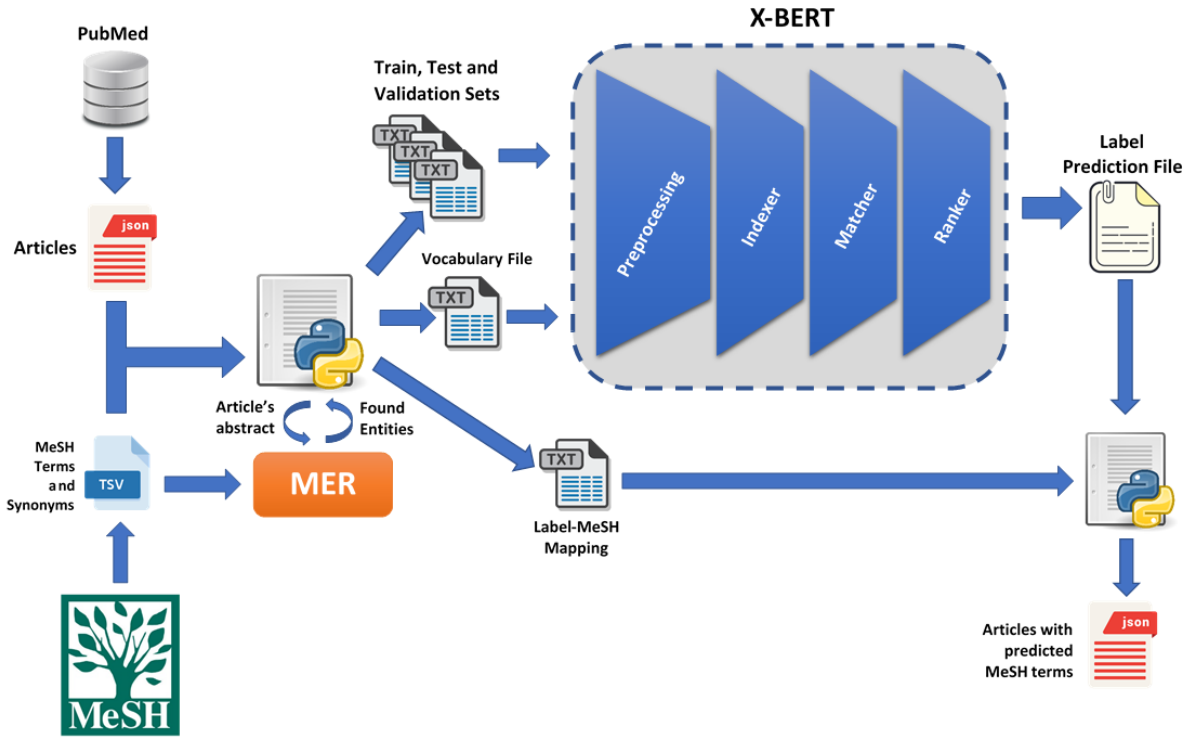

Figure 3.1: Developed pipeline to process the articles before running them through X-BERT and finally converting the resulting predictions into the required format for the competition. This pipeline was used for both BioASQ task 8a and task MESINESP. In MESINESP, instead of using the MeSH terms, the DeCS terms were used.

corresponding direct ancestors as an improvement over the one used for task 8a. This way, additional terms could be recognized and possibly improve the predictions of the algorithm. The terms identified in each abstract were added to the end of the corresponding article title or abstract, depending on the model that was being trained. Repeated terms were removed, and the resulting string was then stemmed using a snowball stemmer from the Python NLTK package[1]. The files were then written, with each line composed by a list of internal numeric identifiers corresponding to the indexed labels, a tab character ('\t') and the stemmed text. This tab character allows X-BERT and X-Transformer to distinguish between labels and text.

After all the files required to run X-BERT or X-Transformer were created, I modified the shell script used to execute the necessary commands to run the different phases of the algorithm, from preprocessing the files, through training the deep learning model and predicting the labels. The modifications made consisted in changing the files used as input for the algorithm, so that it used the previously created files instead of the default files given in the script. Another modification consisted in the usage of CPUs and GPUs and in some parameters that were used during the train of the deep learning model, which will be

[1] https://www.nltk.org/_modules/nltk/stem/snowball.html

referred in detail in further sections of this chapter.

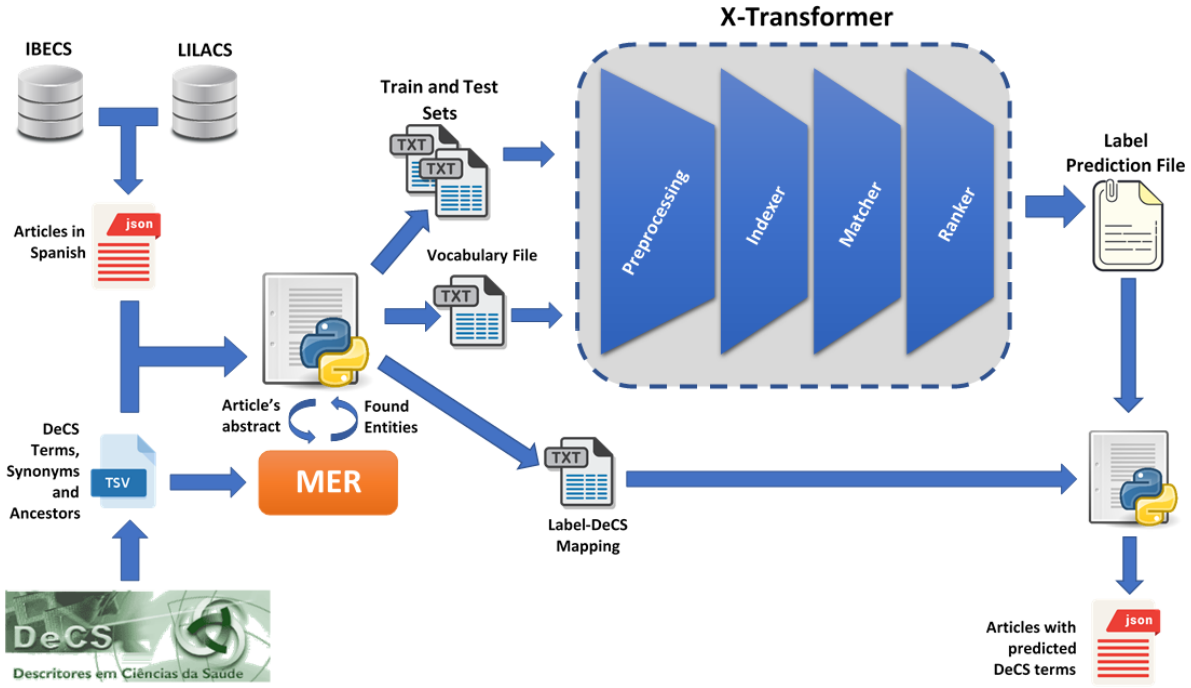

Figure 3.2: Pipeline used in the MESINESP task to process the articles before running them through X-Transformer and finally converting the resulting predictions into the required format for the competition.

The results of the X-BERT and X-Transformer are given in the form of sparse matrices, with a number of rows equal to the number of articles that compose the test set, and the columns corresponding to the possible labels, i.e. the number of MeSH or DeCS terms, depending on the competition. The prediction for each article is retrieved, comprising a top K of most relevant labels and their corresponding confidence values. I used K=20 labels per article for both X-BERT and X-Transformer since, according to the competition organizers, the average number of MeSH/DeCS terms per article in the given datasets was 12.68 for task 8a and 8.12 for task MESINESP. However, for a prediction to be chosen as correct, I discarded every label with a confidence value under a threshold, which will be more detailed in Section 3.4. Each label is then converted to the corresponding DeCS or MeSH term by using the previously created label mapping file, so that in the end, a JSON file is created containing the predicted MeSH or DeCS codes for each article id.

3.3 Data

3.3.1 Training Data

A total of 318,658 abstracts and titles were used as training data in both task 8a and MESINESP. For task 8a, these articles came from a larger dataset with more than 14 million indexed English articles retrieved from PubMed, while for MESINESP, the articles came from a set of 318,658 indexed Spanish written articles retrieved from the IBECS[57] and LILACS[2] databases. The choice of using the same number of articles in both competitions was made so that I could compare the performance between the models developed for English and Spanish.

As was said in the previous section, the articles were split into training, test and validation sets following a 60%-20%-20% proportion to be used by X-BERT, and 70%-30% for X-Transformer. The number of articles used in each set for each model can be seen in Table 3.1.

Table 3.1: Total number of unique DeCS and MeSH term codes and the total number of articles used in the models along with the corresponding proportion used for the train, test and validation sets. In X-BERT, it was 60%-20%-20%. In X-Transformer it was 70%-30%, since there is no validation set.

System	Total Articles	Train Set	Test Set	Validation Set	#MeSH Terms	#DeCS Terms
X-BERT	318,658	191,195	63,732	63,731	29,640	33,702
X-Transformer		223,060	95,598	-		

3.3.2 Competition Test Sets

In the test sets given by competition organizers, there were no MeSH or DeCS terms indexing the articles, so I had to put a placeholder label on each article, because the code of X-BERT and X-Transformer was not prepared to run on unlabeled data. I also had to artificially adapt the size of the given test sets by adding extra articles to them, so that they could have the same size as the test sets used on the previously trained X-BERT or X-Transformer models, as shown previously in Table 3.1. In this case, the set needed to have a total of 63,732 articles for X-BERT models, and 95,598 articles for X-Transformer models.

Since for both competitions the number of articles given to classify was under that value, I have padded the test set with additional articles equal to the difference. For task 8a, I included indexed PubMed articles from the 14 million articles given by the competition as training data. For MESINESP, I included translated and indexed PubMed articles that came from a dataset given as additional material by the task organizers to the participants. The padded articles were used as an additional validation set to define the confidence threshold values of my submissions for both competitions and were not used in neither set to train nor evaluate the models, so that the results would not come biased.

[2] https://lilacs.bvsalud.org/

3.4 Evaluation Script

To evaluate the results achieved by the models developed for the competitions, a Python script was developed by me to measure the Macro and Micro Precision, Recall and F1 scores, which were some of the measures used in both BioASQ competitions. For each model, two evaluations were made: one using the models test set, and another using the additional validation set composed by the articles that were padded to the test sets given by the competition organizers, and which acted as a preliminary evaluation for the competition.

The evaluation script was focused on choosing the confidence score threshold that achieved the highest Micro F1 (MiF) score and the highest Micro Precision (MiP). The focus on MiF was chosen, since MiF was defined by the organization as the most important measure to both competitions. The focus on MiP was motivated by the improved precision of X-BERT and X-Transformer over other XMLC algorithms [17, 15], thus I wanted to see how high it could reach with this type of data.

The confidence score interval given by X-BERT to its predicted labels ranged from 0 to 1. However, in X-Transformer, I noticed that this interval was changed from 0 to 1 to -0.99 to 1, thus encompassing negative confidence values. To facilitate the comparisons between X-Transformer and X-BERT, I normalized this scale to fit the interval from 0 to 1. Then, to determine the best threshold for the confidence score that achieved the highest MiF or MiP values, I tested the values between 0.01 and 1, using increments of 0.01.

3.5 Developed Models

3.5.1 Pre-Competition

3.5.1.1 MER vs No-MER

The first two models trained were used to check if the usage of MER would improve the results of the XMLC algorithm. To test this, one model was trained using MER and the other model was trained without using it. Both models were trained with X-BERT using BERT Base Multilingual Cased and finetuned using the MESINESP abstracts, since the MESINESP dataset was made available earlier. For both models, the parameters given as default by X-BERT to train and evaluate were kept, except for the eval and train batch sizes which were changed from their corresponding default values of 64 and 32 to 3 due to hardware limitations. The number of train epochs was also changed to 7, since the best checkpoint would usually be saved at the end of epoch 6. The models were trained on a single NVIDIA Tesla P4 GPU.

From evaluating the models using the corresponding test set, it is possible to observe in Table 3.2 that the model trained with MER achieved slightly better results than the model that did not use MER, which was expected, since a list of key words related to the article's topics was present, thus improving the algorithm predictions.

Table 3.2: Comparison between the results of two X-BERT models trained with the same datasets and parameters, but one used MER to find DeCS terms, their synonyms and ancestors, and then add them to the end of the corresponding article, while the other did not.
The threshold corresponds to the minimum confidence score used to obtain the highest MiF score.
Bold values correspond to the measures where the usage of MER had increased the results.

Model	Micro			Macro			Threshold
	Precision	Recall	F1	Precision	Recall	F1	
Abstracts	0.4704	0.4101	0.4382	0.4663	0.4022	0.3936	0.08
Abstracts + MER	**0.4780**	**0.4123**	**0.4427**	**0.4766**	**0.4094**	**0.4028**	0.10

3.5.1.2 Titles vs Abstracts

After confirming that the usage of MER improves the results of X-BERT, the next step was to check if a model trained using the titles together with the entities recognized by MER in the corresponding abstracts could achieve better results than a model trained using the abstracts and the entities recognized by MER. The hypothesis was that, due to the increased objectiveness of the titles in the core topics of the article in a less amount of words than the abstract, along with a list of key terms and synonyms given by MER, the model would be able to achieve better results since it had a continuous string of words that were more related to the topics of the article than a larger amount of words with the key topics more diluted. To test the hypothesis, I used X-BERT using the BERT Base Multilingual Cased model and the MESINESP dataset. The parameters used to train the models were the same as the ones described in Section 3.5.1.1.

Table 3.3: Comparison between the results of two X-BERT models, one trained using the abstracts and the other using the titles. Both models were evaluated with a focus on MiF and MiP.
The threshold corresponds to the minimum confidence score used to obtain the highest MiF or MiP score.
Bold values correspond to the measures where the model trained using the titles had achieved better results than its abstract counterpart

Model	Focus	Micro			Macro			Threshold
		Precision	Recall	F1	Precision	Recall	F1	
Abstracts + MER	MiF	0.4780	0.4123	0.4427	0.4766	0.4094	0.4028	0.10
	MiP	0.7117	0.1574	0.2578	0.6812	0.1642	0.2480	0.93
Titles + MER	MiF	0.4685	**0.4142**	0.4397	0.4704	**0.4229**	**0.4072**	0.11
	MiP	**0.7154**	**0.1636**	**0.2663**	**0.6901**	**0.1769**	**0.2617**	0.92

Analyzing the evaluation results in Table 3.3, the difference between using the titles and the abstracts is minimal. If evaluation is focused in achieving the highest MiF value, the model that used the abstracts achieved slightly higher MiP and MiF values. However, if the evaluation was focused on attaining the highest MiP value, the model using the titles achieved better scores in all measures in contrast with its

abstract counterpart, which confirms the hypothesis.

3.5.2 Competition Models

3.5.2.1 Task 8a Models

Only two models were trained for the BioASQ task 8a, with the following characteristics:

- **Model 1** – MER in conjunction with X-BERT, using BERT Base Uncased and finetuned using the article's titles.
- **Model 2** – MER in conjunction with X-BERT, using SciBERT and finetuned using the article's titles.

Both models used the article's titles due to results shown in Section 3.5.1.2. and also due to time restrictions, since the competition had already begun when I started training the models. Therefore, Model 1 was trained for Batch 2 of the competition, while Model 2 was trained for Batch 3. Both models were trained using the same parameters described in Section 3.5.1.1.

3.5.2.2 MESINESP Models

A total of four models were trained for the BioASQ MESINESP task with the following characteristics:

- **Model 1** – MER in conjunction with X-BERT, using BERT Multilingual Base Cased and finetuned using the article's abstracts.
- **Model 2** – MER in conjunction with X-BERT, using BERT Multilingual Base Cased and finetuned using the article's titles.
- **Model 3** – X-Transformer using BERT Multilingual Base Cased and finetuned using the article's abstracts.
- **Model 4** – X-Transformer using BERT Multilingual Base Cased and finetuned using the article's titles.

For this task, I could train four models, since I had more time to submit the results and due to an upgrade in the hardware available. I decided to train two models using X-BERT and the other two using X-Transformer, since it was made available during the time of this competition. The X-Transformer models were supposed to use MER, but due to an error in the scripts, the files did not include the results from MER before being run through X-Transformer. Unfortunately, I could not train new models in time for the competition that included X-Transformer and MER, however, this was done post-competition. I have also decided to train two models using the abstracts and the other two using the titles, so that I

could, not only compare the differences between the two versions of the algorithm, but also check if the hypothesis from Section 3.5.1.2. would also be confirmed in the results achieved in the competition.

The two X-BERT models were trained using the parameters previously described in Section 3.5.1.1. As to the X-Transformer models, the parameters given as default were kept except for the train and eval batch sizes which were both changed from their corresponding values of 64 and 32 to 4. To compensate for the small batch size, the number of gradient accumulation steps was set to 2. The models were trained for 3 epochs, the default value, in a set of 4 GPUs composed by 1 NVIDIA Tesla P4 GPU and 3 NVIDIA Tesla M10 GPUs.

3.6 BioASQ Task 8a Results

3.6.1 Preliminary Evaluation

The results of a preliminary evaluation using the models test set can be seen in Table 3.4. When focused on MiF, both Model 1 and Model 2 achieved similar results for their best scoring threshold values, which was the same. However, Model 2, the one trained using SciBERT, achieved slightly higher values, which was expected since SciBERT was especially created to be used with scientific text. MiP and Macro Precision (MaP) have the highest values, and both Recall and F1 scores have a relatively similar score and are both higher than 0.50 in case of Model 2.

When focused on MiP, the threshold values differ, but the results of both models were again similar, with Model 2 achieving slightly higher values in all measures, except for the Macro and Micro Recall. Both MaP and MiP have increased significantly to values over 0.88. However, this gain in precision comes with the cost of the recall and F1 values dropping between 0.17 and 0.27, approximately.

Table 3.4: Results achieved by the models in the different measures using their corresponding test sets. The values of each measure correspond to the values achieved by the model using the value present in the Threshold column. Bold values correspond to the highest achieved scores in each measure.

Model	Focus	Measures						Threshold
		Micro			Macro			
		Precision	Recall	F1	Precision	Recall	F1	
Model 1	MiF	0.6179	0.4911	0.5472	0.6168	0.4972	0.5293	0.18
	MiP	0.8722	0.2467	0.3846	0.8650	0.2526	0.3695	0.86
Model 2	MiF	0.6270	**0.5095**	**0.5622**	0.6265	**0.5175**	**0.5460**	0.18
	MiP	**0.8894**	0.2408	0.3790	**0.8832**	0.2480	0.3664	0.89

3.6.2 Competition Results

I have competed in batch 2 and 3 of BioASQ task 8a and the models were evaluated in flat and hierarchical measures, as was explained in Section 2.5.2. The achieved results for each measure can be seen in Tables 3.5 and 3.6, where the scores achieved by the X-BERT models are compared with the system that achieved the highest MiF score on the same week.

Table 3.5: Results achieved by the submitted X-BERT models (green lines) compared to the best scoring model in the Micro F-measure (MiF) in the same week. Only the flat measures are presented.
The bold values correspond to values where the model surpassed the best scoring models and reached the 1st place in the corresponding measure.
Measures: MiF – Micro F-measure. MiP – Micro Precision. MiR – Micro Recall. MaF – Macro F-measure. MaP – Macro Precision. MaR – Macro Recall. Acc. – Accuracy. EBP – Example Based Precision. EBR – Example Based Recall. EBF – Example Based F-measure.

Batch	Week	System	Flat Measures									
			MiF	MiP	MiR	MaF	MaP	MaR	Acc.	EBP	EBR	EBF
2	3	deepmesh_dmiip_fdu	0.708	0.758	0.664	0.592	0.721	0.590	0.558	0.755	0.684	0.701
		X-BERT BioASQ (BERT - MiF)	0.316	0.465	0.239	0.066	0.350	0.054	0.184	0.462	0.238	0.294
	4	deepmesh_dmiip_fdu	0.707	0.768	0.655	0.588	0.732	0.584	0.558	0.768	0.673	0.703
		X-BERT BioASQ (BERT - MiF)	0.328	0.496	0.245	0.067	0.352	0.054	0.195	0.498	0.247	0.310
	5	deepmesh_dmiip_fdu_	0.709	0.760	0.663	0.591	0.725	0.591	0.557	0.757	0.678	0.701
		X-BERT BioASQ (BERT - MiF)	0.320	0.477	0.240	0.066	0.343	0.054	0.187	0.480	0.239	0.399
3	2	dmiip_fdu	0.769	0.814	0.730	0.678	0.789	0.674	0.647	0.809	0.744	0.762
		X-BERT BioASQ (SciBERT - MiF)	0.330	0.494	0.248	0.067	0.345	0.055	0.195	0.495	0.247	0.309
	3	dmiip_fdu	0.779	0.830	0.735	0.677	0.810	0.669	0.664	0.825	0.751	0.774
		X-BERT BioASQ (SciBERT - MiP)	0.265	0.803	0.159	0.022	0.640	0.015	0.156	0.796	0.162	0.255
	4	deepmesh_dmiip_fdu	0.710	0.771	0.659	0.594	0.729	0.595	0.563	0.769	0.680	0.706
		X-BERT BioASQ (SciBERT - MiP)	0.245	**0.781**	0.145	0.024	0.639	0.017	0.140	0.750	0.145	0.232
	5	deepmesh_dmiip_fdu	0.706	0.758	0.661	0.589	0.714	0.591	0.556	0.756	0.677	0.700
		X-BERT BioASQ (SciBERT - MiP)	0.260	**0.792**	0.155	0.023	0.618	0.016	0.151	**0.772**	0.157	0.248

In batch 2, the results were submitted for weeks 3, 4 and 5. The submitted solutions were given by Model 1, the model trained using BERT Base uncased, and with the confidence score threshold that achieved the highest MiF score. The results in those weeks were suboptimal, with the model staying in

the last position every week and in all evaluation measures. The values of the Recall measures were the lowest when compared with all other measures, especially Macro Recall (MaR). However, the values in the precision measures were slightly higher than the ones achieved in other measures.

Table 3.6: Results achieved by the submitted X-BERT models (green lines) compared to the best scoring model in the Micro F-measure (MiF) in the same week. Only the hierarchical measures are presented.
Measures: LCA-F – Lowest Common Ancestor F-measure. LCA-P – Lowest Common Ancestor Precision. LCA-R – Lowest Common Ancestor Recall. HiF – Hierarchical F-measure. HiP – Hierarchical Precision. HiR – Hierarchical Recall.

Batch	Week	System	Hierarchical Measures					
			LCA-F	LCA-P	LCA-R	HiF	HiP	HiR
2	3	deepmesh_dmiip_fdu	0.565	0.608	0.550	0.782	0.839	0.762
		X-BERT BioASQ (BERT - MiF)	0.271	0.422	0.218	0.389	0.619	0.316
	4	deepmesh_dmiip_fdu	0.570	0.624	0.546	0.786	0.855	0.752
		X-BERT BioASQ (BERT - MiF)	0.281	0.448	0.221	0.400	0.652	0.319
	5	deepmesh_dmiip_fdu_	0.567	0.615	0.547	0.785	0.844	0.759
		X-BERT BioASQ (BERT - MiF)	0.273	0.434	0.217	0.394	0.641	0.316
3	2	dmiip_fdu	0.657	0.696	0.641	0.829	0.877	0.809
		X-BERT BioASQ (SciBERT - MiF)	0.280	0.445	0.221	0.400	0.651	0.321
	3	dmiip_fdu	0.676	0.715	0.658	0.839	0.889	0.815
		X-BERT BioASQ (SciBERT - MiP)	0.194	0.526	0.125	0.275	0.867	0.175
	4	deepmesh_dmiip_fdu	0.576	0.627	0.555	0.789	0.852	0.760
		X-BERT BioASQ (SciBERT - MiP)	0.182	0.509	0.115	0.251	0.826	0.157
	5	deepmesh_dmiip_fdu	0.568	0.613	0.550	0.784	0.845	0.758
		X-BERT BioASQ (SciBERT - MiP)	0.189	0.510	0.121	0.262	0.840	0.165

In batch 3, Model 2 was used. However, due to hardware constraints, the model was not ready to compete in the first week of this batch. In week 2, Model 2 was submitted following the same strategy used in the previous batch, with the resulting predictions aiming for the highest MiF score. Although, the results achieved by Model 2 were slightly higher than the ones achieved by Model 1, they were still low when compared with the other participants.

In the following weeks, I changed the evaluation focus from MiF to MiP, since it was the best scoring measure in the previous submissions as well as the best scoring measure in the preliminary evaluation. As

a result, the model achieved lower results in the MiF measure as expected, although this loss was less than 0.1. The Recall measures also suffered losses, especially MaR which was the worst measure, dropping to values under 0.02. However, all precision measures increased significantly, making the model reach the top positions in most precision measures. In the MiP measure, Model 2 reached a 3rd, 1st and again 1st place on weeks 3, 4 and 5, respectively. In Example Based Precision (EBP), Model 2 reached the 3rd place in week 4 and the 1st place in week 5. The improvements were also noticed in the Hierarchical Measures, namely in Hierarchical Precision (HiP), in which the model achieved a 3rd place in week 3, and 5th place in weeks 4 and 5. The only precision measures where Model 2 did not have a significant increase were the MaP measure, where the model achieved a 6th place in the last three weeks, and the LCA-P measure, where the model could not surpass more than 3 other competing systems. In the end, the accuracy also dropped by at least 0.04 when compared with the results focused on MiF.

3.7 BioASQ Task MESINESP Results

3.7.1 Preliminary Evaluation

Contrarily to the task 8a preliminary evaluation, for task MESINESP I added a new evaluation parameter, which was the confidence score threshold of 0.50. I decided to add this value since it was the middle of the confidence interval scale and it should provide a baseline performance for the models.

Table 3.7: Results achieved by the models in the different measures using their corresponding test sets. The values of each measure correspond to the values achieved by the model using the value present in the Threshold Value column.
Bold values correspond to the highest achieved scores in each measure.

Model	Focus	Measures						Threshold
		Micro			Macro			
		Precision	Recall	F1	Precision	Recall	F1	
Model 1	Base	0.6570	0.2622	0.3748	0.6295	0.2620	0.3386	0.50
	MiF	0.4780	0.4123	0.4427	0.4766	0.4094	0.4028	0.10
	MiP	0.7117	0.1574	0.2578	0.6812	0.1642	0.2480	0.93
Model 2	Base	0.6492	0.2663	0.3777	0.6307	0.2771	0.3512	0.50
	MiF	0.4685	0.4142	0.4397	0.4704	0.4229	0.4072	0.11
	MiP	0.7154	0.1636	0.2663	0.6901	0.1769	0.2617	0.92
Model 3	Base	0.7026	0.2819	0.4024	0.6890	0.2922	0.3773	0.50
	MiF	0.4826	**0.4971**	**0.4898**	0.4830	**0.5046**	**0.4587**	0.24
	MiP	0.7249	0.2539	0.3761	0.7081	0.2652	0.3555	0.55
Model 4	Base	0.6928	0.2785	0.3974	0.6836	0.2984	0.3798	0.50
	MiF	0.5137	0.4318	0.4693	0.5140	0.4513	0.4437	0.28
	MiP	**0.7765**	0.1465	0.2465	**0.7595**	0.1674	0.2567	0.79

The results achieved in the first evaluation using the models test set can be seen in Table 3.7. For both Model 1 and Model 2, the scores achieved by each model were very similar, with slightly higher scores achieved by Model 2. As for Models 3 and 4, which used X-Transformer, there is a clear improvement in the results when compared with the models that used X-BERT, even without the usage of MER. Also, and contrarily to Models 1 and 2, Model 3, which used the abstracts, achieved higher values in all measures when compared with Model 4, except when focused on MiP where Model 4 surpassed it.

In the second evaluation, which used the additional validation set, the results achieved were different as can be seen in Table 3.8. First, there was a drop in the scores, especially in the recall measures thus dropping the F1-score. Precision also dropped, but not as much as the other measures. When focusing on MiP, the best performing model was Model 2, which achieved higher scores when compared with Model 4. This small difference might be due to the usage of MER in Model 2, while Model 4 did not. As to the focus on MiF, the best scoring model was Model 4 which achieved superior recall and F1-score values when compared with the other models.

Table 3.8: Results achieved by the models when using the test set composed by additional indexed articles from PubMed translated to Spanish.
Bold values correspond to the highest achieved values in the corresponding measures.
Green lines correspond to the models which predictions were submitted to the MESINESP challenge.

Model	**Focus**	**Measures**						**Threshold**
		Micro			**Macro**			
		Precision	**Recall**	**F1**	**Precision**	**Recall**	**F1**	
Model 1	Base	0.6132	0.0787	0.1395	0.6117	0.0838	0.1400	0.50
	MiF	0.2937	0.1506	0.1991	0.3163	0.1545	0.1898	0.05
	MiP	0.6737	0.0550	0.1018	0.6627	0.0603	0.1082	0.97
Model 2	Base	0.6290	0.0759	0.1355	0.6323	0.0814	0.1377	0.50
	MiF	0.3029	0.1572	0.2070	0.3204	0.1608	0.1975	0.05
	MiP	**0.6903**	0.0556	0.1028	**0.6823**	0.0613	0.1103	0.95
Model 3	Base	0.5610	0.0824	0.1437	0.5848	0.0886	0.1437	0.50
	MiF	0.3198	0.1730	0.2245	0.3314	0.1797	0.2143	0.15
	MiP	0.6468	0.0489	0.0910	0.6471	0.0550	0.0998	0.99
Model 4	Base	0.6448	0.0832	0.1473	0.6374	0.0896	0.1499	0.50
	MiF	0.3786	**0.2041**	**0.2652**	0.3761	**0.2107**	**0.2506**	0.14
	MiP	0.6823	0.0549	0.1017	0.6752	0.0613	0.1103	0.77

After this second evaluation, it was necessary to choose which models should have their predictions submitted for the MESINESP task, for which I could submit up to four models. Considering the results of the second evaluation, I chose to submit the results from a model that used X-BERT and from a model that used X-Transformer, so that the performance of both could be compared in the competition, therefore, Model 2 and Model 4 were chosen. From Model 2, the results from the focus on MiF and MiP were

submitted, since it had the highest achieved MiF score in the models that used X-BERT, and since the focus on MiP had achieved the highest precision scores in the evaluation. From Model 4, the results from the baseline and the focus on MiF were chosen, since the focus on MiF achieved the highest MiF and recall values from the evaluation, and the baseline because the precision gains of the MiP focus were not so significant and had heavier losses in the other measures when compared with the predictions using the baseline.

3.7.2 Competition Results

Like in BioASQ task 8a, the results in the MiF measure were not optimal, with the four submissions staying in bottom positions in the classification. However, in the precision measures, the scores achieved by two of my submissions were higher than most systems, even surpassing the winning system.

Table 3.9: Results achieved by the models submitted for the MESINESP task (green lines) compared to the best scoring model in the Micro F-measure (MiF) and the BioASQ Baseline.
Bold values correspond to the measures where my submissions surpassed both the BioASQ baseline and the winning system.
Measures: MiF – Micro F-measure. MiP – Micro Precision. MiR – Micro Recall. MaF – Macro F-measure. MaP – Macro Precision. MaR – Macro Recall. Acc. – Accuracy. EBP – Example Based Precision. EBR – Example Based Recall. EBF – Example Based F-measure.

System	Flat Measures									
	MiF	**MiP**	**MiR**	**MaF**	**MaP**	**MaR**	**Acc.**	**EBP**	**EBR**	**EBF**
Winning System	0.4254	0.4374	0.4140	0.3194	0.3989	0.3380	0.2786	0.4382	0.4343	0.4240
BioASQ Baseline	0.2695	0.2337	0.3182	0.2816	0.3733	0.3220	0.1659	0.2681	0.3239	0.2754
X-BERT BioASQ F1 (Model 2 focus MiF)	0.1430	**0.4577**	0.0847	0.0220	0.3095	0.0186	0.0787	**0.5057**	0.0867	0.1397
X-BERT BioASQ (Model 2 focus MiP)	0.0909	**0.5449**	0.0496	0.0045	0.3422	0.0036	0.0503	**0.5415**	0.0508	0.0916
LasigeBioTM TXMC F1 (Model 4 focus MiF)	0.2507	0.3559	0.1936	0.0858	0.3646	0.0799	0.1440	0.3641	0.1986	0.2380
LasigeBioTM TXMC P (Model 4 focus baseline)	0.1271	**0.6864**	0.0701	0.0104	**0.6989**	0.0081	0.0708	**0.6609**	0.0716	0.1261

The predictions of Model 4 focused on the baseline achieved the 2nd place in the MiP, MaP and EBP measures and also the highest accuracy from my submissions, while the predictions of Model 2 focused on MiP achieved the 3rd place in MiP as well as a 5th place in EBP. As to the models focused of MiF, they achieved balanced scores between all measures, but they were not enough to surpass the competition baseline. It is also noticeable that the submissions focused on MiF achieved higher accuracy scores. The results achieved by the four submissions, the BioASQ baseline and the winning system are presented in Table 3.9.

3.8 Discussion

3.8.1 Achieved Results

The results achieved in both these competitions show that the usage of deep learning XMLC solutions in the biomedical and multilingual panorama can achieve promising results, especially with the high MiP values achieved in MESINESP and in the last three weeks of task 8a. However, there is also a great room for improvement. For example, in both task 8a and task MESINESP, the models struggled in all recall measures, with scores lower than most of the competing systems. In contrast, the models achieve higher scores in the precision measures, especially when the submitted results were focused on MiP. However, the focus on MiP came with significant losses on recall, thus leading to lower F-measure scores and lower accuracy. These low recall values can possibly be explained by several factors.

The amount of data used to train the models could have impacted the results. I have chosen to use the same number of articles in both competitions, which was the number of available articles in the pre-processed train set from MESINESP, so that I could compare the performance of a biomedical XMLC based model across multiple languages. There was much more data available for both task 8a and task MESINESP that I could have used to train larger models. However, this was not possible to do in time due to the hardware limitations.

Another limitation, namely in task 8a and in the X-Transformer models, was the fact that I did not use more complex lexicons in MER to identify entities in the text. The lexicon used in task 8a was rather simple, only comprising the MeSH terms and their synonyms. I could have used more complex lexicons such as one with the term's ancestors or with ontology information, which could possibly increase the final scores. In task MESINESP, the fact that I did not use MER in the X-Transformer models could have negatively influenced results.

Finally, another reason for the low recall values achieved could be caused by the fact that XMLC algorithms tend to achieve higher precision values due to an increased focus on precision measures, namely in Precision at top K (P@K). P@K is widely used in the evaluation and comparison between XMLC algorithms [13, 17, 15, 80, 81, 82, 14, 79], as it considers the number of most relevant labels retrieved by the algorithm among the top K documents. The choice of using this measure to evaluate the performance of these algorithms makes sense when dealing with datasets with thousands of documents, from which the objective is to retrieve the most relevant labels from a set of labels that can reach millions. However, this increased focus in precision can affect negatively the evaluation of the systems in other measures, such as the F-score and recall, which were evaluated in these two competitions.

3.8.2 English vs Multilingual

One of my objectives with the participation in these two competitions was to compare the performance of an English XMLC model versus its multilingual counterpart in a biomedical domain. When comparing the results achieved by my submissions in both competitions, it is possible to notice that there is a great

difference between the performances of the English models and the multilingual models.

In task 8a, Models 1 and 2 shared the same characteristics as Model 2 that was submitted for task MESINESP. The three of them were X-BERT models trained with the same number of articles, all used the article's titles and were combined with MER to find key terms in the article's abstracts. The major difference between them was the pre-trained language models used. In task 8a, the models used BERT Base Uncased and SciBERT, while in task MESINESP the model used BERT Base Multilingual Cased.

Analyzing the scores achieved by these three models, it is possible to notice a significant difference in the scores achieved in my evaluations and in the competition evaluations. The clearest example is in BioASQ task 8a evaluation, where the worst score achieved by my submissions in the competition surpasses my best scored submission in task MESINESP in the non-precision measures, as well as the enormous difference in the scores between the first classified systems in task 8a and the MESINESP winner. One reason for this difference can be caused by the number of documents to classify in each task, which in MESINESP was more than 24,000 articles, whereas in task 8a was about 5,000 to 7,000 articles in each week.

Table 3.10: Results of the different versions of the common Model (Article's titles + terms identified by MER) used in the BioASQ competitions, when evaluated using the corresponding test set.
Bold values correspond to the highest values achieved in the corresponding measures.

Model	Focus	Measures						Threshold
		Micro			Macro			
		Precision	Recall	F1	Precision	Recall	F1	
BERT Base Uncased	MiF	0.6179	0.4911	0.5472	0.6168	0.4973	0.5293	0.18
	MiP.	0.8722	0.2467	0.3846	0.8650	0.2526	0.3695	0.86
SciBERT	MiF	0.6270	**0.5095**	**0.5622**	0.6265	**0.5175**	**0.5460**	0.18
	MiP.	**0.8894**	0.2408	0.3790	**0.8832**	0.2480	0.3665	0.89
BERT Multilingual Base Cased (MESINESP)	MiF	0.4685	0.4142	0.4397	0.4704	0.4229	0.4072	0.11
	MiP	0.7154	0.1636	0.2663	0.6901	0.1769	0.2617	0.92

However, in the preliminary evaluations that used the model's test sets, the difference is also evident and show that the English-based models achieve better results in all measures, as it can be seen in Table 3.10. The reason for this difference in the results can be caused by the amount of data used to train the pre-trained language models. The BERT models were trained using mostly text data retrieved from Wikipedia. However, the amount of Wikipedia text varies between different languages. For example, in terms of words, the BERT base model was trained with more than 3,000 million words, with about 2,500 million coming from the English Wikipedia [5]. In comparison, the number of words in the Spanish Wikipedia, as of June 2020, was less than 900 million words. This enormous gap between the number of words is very significant and consequently reflects on the results achieved by the non-English models.

3.9 Post-Competition

3.9.1 X-Transformer and MER

Due to a small error in the script, the entities identified by MER were not included in the text before being given as input to X-Transformer. After correcting the error, I executed the procedure described earlier in Section 3.2. using the article's titles and parameters used for the MESINESP competition, along with the lexicon composed by the DeCS terms, their synonyms and direct ancestors.

The results achieved using the model's test set can be seen in Table 3.11 and confirm what was expected and that has been previously seen with X-BERT in Section 3.5.1.1. The model that used MER achieved slightly higher results in all measures and in all evaluation focuses. The only exceptions occur in the evaluation focused on MiF, where the MiP score achieved by the model that used MER is slightly lower, and when the evaluation uses the baseline threshold, where MaR has a minimal difference of 0.001. The greatest improvements in the evaluation measures are achieved when the evaluation is focused on MiF, where MiR, MiF, MaR and MaF all present values with an improvement greater than 0.01.

Table 3.11: Comparison between two X-Transformer models (Article's titles) using the corresponding test sets, but one model used MER to identify DeCS terms, their synonyms and ancestors in the article's abstract before executing X-Transformer.
Bold values correspond to the highest values achieved in the corresponding measures.

Model	Focus	Measures						Threshold
		Micro			Macro			
		Precision	Recall	F1	Precision	Recall	F1	
X-Transformer Titles	Base	0.6928	0.2785	0.3973	0.6836	0.2984	0.3798	0
	MiF	0.5137	0.4319	0.4693	0.5140	0.4513	0.4437	0.28
	MiP	0.7766	0.1465	0.2465	0.7595	0.1674	0.2567	0.79
X-Transformer Titles + MER	Base	0.6994	0.2792	0.3991	0.6903	0.2973	0.3814	0
	MiF	0.5087	**0.4623**	**0.4844**	0.5116	**0.4776**	**0.4569**	0.27
	MiP	**0.7810**	0.1496	0.2512	**0.7638**	0.1701	0.2610	0.78

3.9.2 Different Amounts of Data

The low recall and F1-score values achieved in both BioASQ task 8a and MESINESP could be related with the amount of data used to create the models. In both competitions, I could have used more data to train and evaluate the models but instead, I opted to use the same number of articles for both competitions so that I could compare the performance between the English and Spanish XMLC models. Then, due to time and hardware constrains, I could not train larger models in time for the competitions.

In order to test how the amount of data affected the results, I decided to train new models using additional datasets created with the data given in the competitions. The goal was to train new X-BERT and X-Transformer models using different amounts of articles, and then compare the results of these new models with the results achieved by the models used on the competitions. However, due to hardware constrains, I could not retrieve the predictions of the larger models of X-Transformer. A possible way of countering this limitation would be reducing the number of predicted labels per article, but I opted not to do that, since this change could affect the scores achieved by the model in the evaluation, and thus the comparison with the other models could lead to inaccurate conclusions. Therefore, the test was only made using X-BERT models and using the data given in BioASQ task 8a.

From the articles given by the competition organizers for task 8a, I created one dataset with half the size of the one used in the competition and another with double that size. The distribution of the articles in the train, test and validation sets followed the same proportion of 60%-20%-20% that was used by the X-BERT models trained for the competition. The number of articles present in each dataset can be seen in Table 3.12.

Table 3.12: Size of the datasets created to compare the performance of X-BERT models using different amounts of data. The English Set corresponds to the existing dataset that I used in the BioASQ task 8a competition. English Set S has half the size of English Set, while English Set L has double the size of English Set. The proportion used for the train, test and validation sets was 60%-20%-20%.

Dataset	Total Articles	Train Set	Test Set	Validation Set	#MeSH Terms
English Set S	159,329	95,598	31,866	31,865	29,640
English Set	318,658	191,195	63,732	63,731	
English Set L	637,316	382,389	127,463	127,464	

Then, following the pipeline described in Section 3.2 for BioASQ task 8a, two models were created, one for each newly created dataset, with the following characteristics:

- **Model S** – MER in conjunction with X-BERT, using SciBERT and finetuned using the article's titles from English Set S.
- **Model L** – MER in conjunction with X-BERT, using SciBERT and finetuned using the article's titles from English Set L.

The two models would be compared with Model 2 described in Section 3.5.2.1, which will be called Competition Model. The parameters used to train the new models were the same as the ones that were previously described in Section 3.5.1.1. This choice was made so that the training conditions could be equal for each model, with the only difference being the amount of data used to train and evaluate the models.

Looking at the results achieved by the models in Table 3.13, it is possible to notice considerable differences with the usage of smaller and larger quantities of data. Model S achieved higher precision values than the Competition Model when the evaluation was focused on MiP. However, it presents worst results in all other measures, especially in the recall measures. Interestingly, the highest recall and F1-score values are achieved by the Competition Model. In contrast, Model L achieved higher precision values, except when the evaluation focus was on MiF.

I was expecting superior values in all measures for Model L when compared with the other two models since it had more data available to learn from, but the Competition Model achieved higher recall and F1-score values. The reason for these results could lie in the training parameters, namely in the number of train epochs. The models were all trained during 7 epochs, which was the ideal number of epochs for the Competition Model. However, this number could have different for the other two models. In fact, when I analysed the X-BERT models training logs, the best checkpoint for Model S occurs in epoch 4, while the best checkpoint for Model L occurs at the end of epoch 7 which means that, for Model L, the real best scoring checkpoint might not have been achieved, thus the results could have been better. The only way to confirm this hypothesis would be training Model L with an increased number of epochs, which will be left as future work.

Table 3.13: Results achieved by the X-BERT models in the corresponding test sets using different size datasets in the English language.
Bold values correspond to the highest achieved values in the corresponding measure.

Model	**Focus**	**Measures**						**Threshold**
		Micro			**Macro**			
		Precision	**Recall**	**F1**	**Precision**	**Recall**	**F1**	
Model S	Base	0.7873	0.3477	0.4823	0.7853	0.3520	0.4603	0.50
	MiF	0.6058	0.4971	0.5461	0.6057	0.5037	0.5286	0.16
	MiP	0.9125	0.1662	0.2811	0.9020	0.1726	0.2753	0.99
Competition Model	Base	0.7880	0.3773	0.5103	0.7861	0.3840	0.4914	0.50
	MiF	0.6270	**0.5095**	**0.5622**	0.6265	**0.5175**	**0.5460**	0.18
	MiP	0.8894	0.2408	0.3790	0.8832	0.2480	0.3664	0.89
Model L	Base	0.8034	0.3504	0.4879	0.8013	0.3553	0.4672	0.50
	MiF	0.6208	0.5038	0.5562	0.6207	0.5107	0.5391	0.16
	MiP	**0.9324**	0.1551	0.2660	**0.9194**	0.1624	0.2636	0.99

Chapter 4

Clinical Case Classification – CANTEMIST Competition

The usage of text mining and NLP techniques in clinical cases can be crucial to retrieve key pieces of information about pathologies, such as symptoms or treatments, increasing the available knowledge about the pathology, which can improve the research and clinical decision-making [83, 84]. However, there are not so many tools developed for the multilingual panorama, being most tools focused on the English language, or with tools not specialized for processing clinical cases [84].

Taken this into account, I decided to participate in the CANTEMIST international challenge, namely in the CANTEMIST-CODING task. In this task, the goal was the classification of Oncology clinical cases written in Spanish by returning a list of eCIE-O codes related to the content of each clinical case, ranked by the order of confidence. Since it was a multi-label classification challenge, I decided to apply and adapt the X-Transformer pipeline that was used for the BioASQ MESINESP task, since it had achieved promising results in the precision measures of that challenge, and this way it could also serve as an additional case study of this XMLC solution to the multilingual biomedical panorama.

In this chapter, I will start by explaining the inclusion of a new pre-trained language model to train X-Transformer. Then, in Section 4.2 I will explain the developed pipeline used in the CANTEMIST-CODING task and in Section 4.4 the developed models for the task. The evaluation and the results achieved in the challenge are tackled in Sections 4.5 and 4.6, correspondingly, finally followed by a discussion in Section 4.7.

4.1 X-Transformer modifications

The only modification made in X-Transformer for this challenge, in addition to the ones that were already made for the BioASQ competitions and that were described in Section 3.1, was the inclusion of BETO[1], the Spanish version of BERT, in the choices of models to train X-Transformer. This inclusion was made

[1] https://github.com/dccuchile/beto

since I considered that using a BERT model specifically designed for the Spanish language could lead to improved results over the Multilingual version of BERT, as it can be seen in the evaluation of BETO [85].

4.2 Pipeline

Like in the BioASQ competitions, I developed a pipeline to execute the X-Transformer algorithm and retrieve its predictions in the required format by the CANTEMIST-CODING task. This pipeline can be seen in Figure 4.1 and it was based in the pipeline developed for the BioASQ competitions. However, and contrarily to the BioASQ competitions, it does not use MER to identify key entities in the text due to the size of the texts that composed the clinical texts, which largely surpassed the 512 token limit of BERT [5], therefore, adding extra text to the clinical cases may not improve the results.

The first step was retrieving the data from the competition organisers. The data was organized in separate folders for the training set and development sets, with each folder containing the text files of each clinical case along with '.ann' files that contained the eCIE-O terms and codes that were indexed to each clinical case.

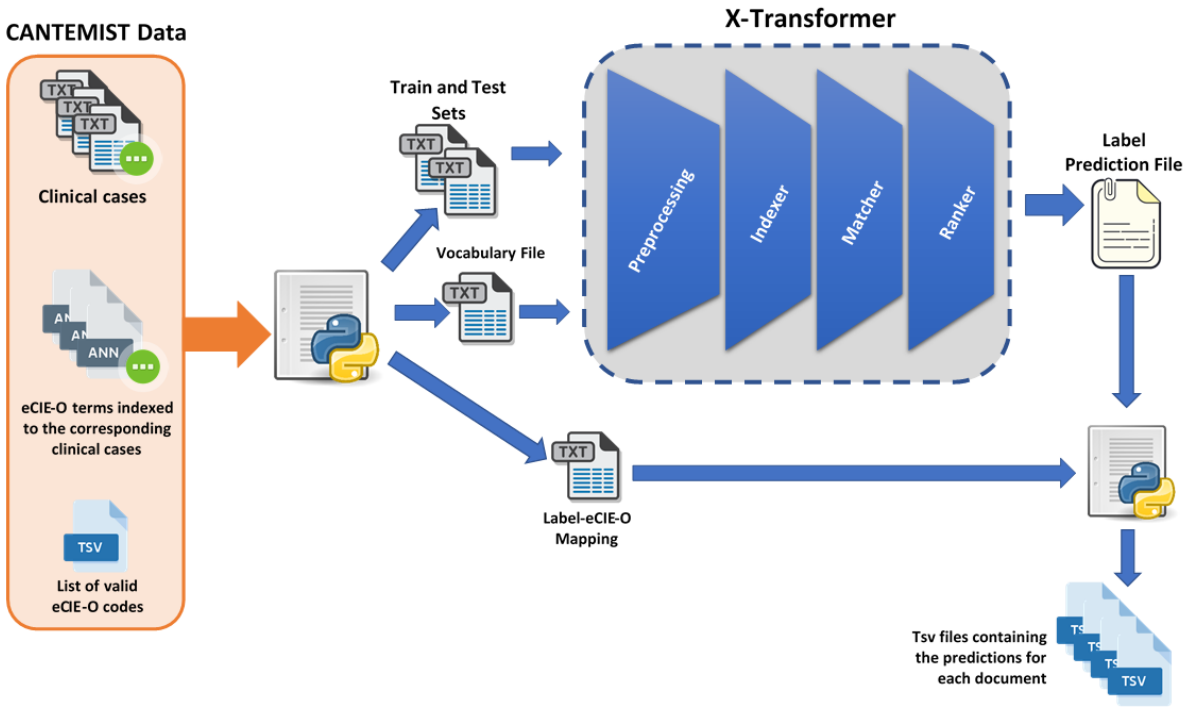

Figure 4.1: Developed pipeline for the CANTEMIST-CODING task. It processes the CANTEMIST data before running it through X-Transformer. In the end, the predictions are converted to the format required by the competition.

After retrieving the data, the next step was merging each separate text file that composed the training and development datasets into a single file for each dataset, so that in the end I would have only two files, one containing the clinical cases that would be used as train set and the other the clinical cases that would be used as the test set. Then, using the '.ann' files that were associated with each clinical case, I extracted all labels that were attributed to each document and appended them to the beginning of the corresponding clinical case separated by a tab character ('\t'), so that X-Transformer could distinguish between the labels and the text. The text was then stemmed using a Snowball stemmer[2].

The next step was the creation of the vocabulary file that contained the list of valid labels. Similar to the vocabulary files of the BioASQ tasks, each line of this file had an eCIE-O code and its corresponding internal identifier, corresponding to a number from 0 to N, where N is the total number of eCIE-O codes minus 1. For the creation of this file, I used a file containing a list of valid eCIE-O codes that was provided by the competition, from which I retrieved all codes and corresponding descriptions. Then, I included the eCIE-O code descriptions that were present in the '.ann' files and that were not present in the list previously retrieved. The codes without any description were removed, since they would be of no use to classify the clinical cases using X-Transformer. In the end, the vocabulary file was composed by a total of 4360 eCIE-O codes. Then, I have also created a label mapping file that contained the correspondence between each eCIE-O code and its numeric identifier in the vocabulary file, which would later be used to map the predictions from their numeric identifiers to the corresponding eCIE-O codes, similar to what was described in the previous chapter.

The sparse matrices containing the predictions for each clinical case were retrieved, comprising a top K of most relevant labels, with K=20, and their confidence values ranging from -0.99 to 1. Then, using a Python script, the predicted labels were converted from their numeric identifiers to their corresponding eCIE-O codes using the label mapping file previously created. The script also discarded each label with a confidence score under a threshold chosen to achieve the highest Precision, Recall or F1 scores. For each of these measures, a '.tsv' file was created with the predictions for each clinical case in the format required by the competition. A fourth '.tsv' file was also created for the score threshold equal to 0, which was used as a baseline score.

4.3 Data

4.3.1 Training Data

The data given for the CANTEMIST-CODING task was composed by one training set and two development sets named dev-set1 and dev-set2, correspondingly. The training set was composed by 501 clinical cases and the two development sets had 250 clinical cases each. However, dev-set1 had one unlabeled clinical case which had to be discarded since it would be of no use to train or evaluate a model. Therefore,

[2] https://www.nltk.org/_modules/nltk/stem/snowball.html

dev-set2 was used as the test set to develop the X-Transformer models. With this, the train and test sets followed a proportion of approximately 77%-33% which was used to train the X-Transformer models.

Then, I decided to generate a larger training set composed by the 501 clinical cases and the remaining 249 clinical cases from dev-set1 that were classified with eCIE-O codes. This larger training set would be used to train larger versions of the models so that I could compare how this increment in the data could affect the final results. The models trained with this dataset kept the same test set previously described, composed by the 250 clinical cases from dev-set2. A summary of the datasets can be seen in Table 4.1.

Table 4.1: Total number of clinical cases given by the CANTEMIST competition organizers to develop the models along with the corresponding number of valid eCIE-O terms that had an associated description.
The test set was composed by the clinical cases from dev-set2, while the larger train set was composed by the clinical cases that composed the train set plus the 249 annotated clinical cases that composed the dev-set1.

Total Clinical Cases	Train Set	Larger Train Set	Test Set	#eCIE-O Terms
1000	501	750	250	4360

4.3.2 Competition Test and Background Sets

In the CANTEMIST challenge, the goal was to classify, not only the 300 clinical cases that composed the test set, but also a larger background set composed by 5032 clinical cases. Like in the BioASQ competitions, I had to put a placeholder label on each document, since there were no eCIE-O codes indexing the clinical cases and because X-Transformer was not prepared to run on unlabelled data.

The size of the test set was also artificially adapted so that it could have the same size of the test set used to train the X-Transformer models. However, and contrarily to what happened in the BioASQ competitions, the number of clinical cases that composed the test and background sets was far greater than the number of clinical cases that was used in the models test set, which only had 250 documents. The solution was splitting the test and background sets into a set of smaller files of 250 clinical cases each. The first 109 lines of each of those files were composed by clinical cases from the test and background sets to classify, while the remaining 141 came from dev-set1, which was already classified with eCIE-O codes. These 141 clinical cases were used as an additional validation set to define the confidence threshold values of my submissions, has it was done in the BioASQ competitions.

4.4 Developed Models

In a first iteration, I trained four X-Transformer models using the training set composed by 501 clinical cases. Two models were trained with pre-trained BERT models, namely BERT Base Multilingual Cased and BETO, the Spanish version of BERT. The other two models were trained with two pre-trained models using data from the MESINESP Competition. One of them, used the model described in Section 3.9.1 that

consisted in using the biomedical article's titles along with MER to identify DeCS terms, their synonyms and ancestors in the article's abstracts. The other model used the X-Transformer model described in Section 3.9.2 that was developed with larger amounts of data, but from which I could not retrieve its predictions due to hardware limitations. However, since the model files were created, the model could be used to train other X-Transformer models. These two models shall be referred as MESINESP Model and MESINESP Model Large, since they were trained with data from that competition. Summarizing, the four models had the following characteristics:

- **Model 1** - BERT base Multilingual Cased finetuned with 501 clinical records.
- **Model 2** - BETO finetuned with 501 clinical records.
- **Model 3** - MESINESP Model finetuned with 501 clinical records.
- **Model 4** - MESINESP Model Large finetuned with 501 clinical records.

In a second iteration, I trained four additional models following the same characteristics of the previous ones, but using the larger train set composed by the 750 clinical cases. This way, I could compare the results of the models trained with different amounts of clinical cases. The models had the following characteristics:

- **Model 5** - BERT base Multilingual Cased finetuned with 750 clinical records.
- **Model 6** - BETO finetuned with 750 clinical records.
- **Model 7** - MESINESP Model finetuned with 750 clinical records.
- **Model 8** - MESINESP Model Large finetuned with 750 clinical records.

All models were trained using the default parameters of X-Transformer, except for the eval and train batch sizes which were both changed from their original values of 64 and 32 to 4 due to hardware constrains. The number of gradient accumulation steps was set to 2 to compensate for the small batch size and each model was trained for 12 epochs, on a single NVIDIA Tesla P4 GPU.

4.5 Preliminary Results

In order to choose which model predictions to submit, I decided to evaluate the predictions made by each model on the dev-set2 using the evaluation script given by the competition organizers. As was explained before, each model had four '.tsv' files as output, with each file corresponding to the confidence score threshold that achieved the best precision, recall, F1-score and the baseline score, which corresponded to the confidence score threshold set to 0, which was the middle of the X-Transformer confidence score scale. Then, each file was used as input for the evaluation script given by the competition organizers.

Table 4.2: Preliminary results of the trained models for the CANTEMIST-CODING task using the second development set.
Bold values correspond to the three highest values achieved in the Mean Average Precision (MAP) measure.
Green lines correspond to the models and corresponding evaluation focus whose predictions were chosen to submit for the CANTEMIST-CODING task.

Model	Focus	MAP	Threshold
Model 1 BERT Base Multilingual Cased	Baseline	0.222	0
	F1	0.366	-0.66
	Precision	0.180	0.22
	Recall	0.384	-0.81
Model 2 BETO	Baseline	0.267	0
	F1	0.385	-0.48
	Precision	0.188	0.48
	Recall	0.438	-0.83
Model 3 MESINESP Model	Baseline	0.293	0
	F1	0.378	-0.34
	Precision	0.191	0.52
	Recall	**0.446**	-0.82
Model 4 MESINESP Model Large	Baseline	0.281	0
	F1	0.396	-0.45
	Precision	0.180	0.60
	Recall	**0.448**	-0.82
Model 5 BERT Base Multilingual Cased (750 CC)	Baseline	0.203	0
	F1	0.372	-0.46
	Precision	0.186	0.12
	Recall	0.407	-0.83
Model 6 BETO (750 CC)	Baseline	0.224	0
	F1	0.371	-0.46
	Precision	0.180	0.39
	Recall	0.417	-0.82
Model 7 MESINESP Model (750 CC)	Baseline	0.250	0
	F1	0.392	-0.40
	Precision	0.200	0.32
	Recall	**0.442**	-0.82
Model 8 MESINESP Model Large (750 CC)	Baseline	0.268	0
	F1	0.379	-0.43
	Precision	0.194	0.29
	Recall	0.427	-0.81

The results for each model are registered in Table 4.2, with the highest MAP scores being achieved when the predictions are focused on achieving the highest recall scores. Another fact to notice is that the highest MAP scores are achieved by the X-Transformer models trained with the MESINESP models,

which was expected, since the MESINESP data consisted in biomedical articles written in Spanish. It is also noticeable that the models that use BETO seem to achieve higher MAP scores when compared with the ones that used BERT Multilingual. As to the usage of more clinical cases to train the models, there is not a clear evidence that it improves the results achieved, with some models achieving slightly higher scores in MAP if the evaluation was focused on precision or in the F1-score, while in other models the score was inferior when compared with the models that used lesser articles.

Taken these results into consideration, I decided to choose 5 distinct models to submit, with the predictions focusing on recall, since it was the measure that lead to increased MAP values. The chosen models were Models 2, 3, 4, 5 and 7.

The predictions made by the models for each of the 48 text files that contained the data from the test and background sets were retrieved, resulting in 48 prediction files. For each of those prediction files, the first 109 lines, which corresponded to the predictions made for the test and background sets, were stored in a variable which contained the data to be written in the '.tsv' files. The remaining 141 lines, which corresponded to the predictions of the classified clinical cases from dev-set1, were used to find the confidence score threshold that achieved the best recall score, so that it could be used to discard the test and background sets predictions with a confidence score under that value before writing the '.tsv' file to be submitted.

4.6 Competition Results

The submissions for CANTEMIST-CODING task consisted in the predictions for both the test and background sets. However, the evaluation only considers the 300 clinical cases that compose the test set. The submissions were evaluated on the MAP measure, which was the most relevant measure for this challenge, but the precision, recall and F1 scores were also evaluated. The evaluation also made distinction between considering the eCIE-O term 'Metastasis' and not considering it, since the dataset presented an high class imbalance in which mentions to this term are ubiquitous.

Looking at results achieved in the competition presented in Table 4.3, the best scoring model was Model 5 contrarily to what I was expecting, since in the preliminary evaluation this model had achieved the lowest MAP score of the five models submitted for this task. Interestingly, the lowest MAP scores were achieved by Models 3, 4 and 7, which were trained using the MESINESP models and that had achieved the highest MAP scores in the preliminary evaluation. As to remaining evaluation measures, the precision and F1 scores were lower than the recall scores as expected, since the submissions contained the predictions that used a confidence score threshold focused on achieving the highest recall scores.

Table 4.3: Results of the submitted models for the CANTEMIST-CODING task. The measures without Metastasis are calculated without considering the mentions to the eCIE-O term 'Metastasis' (8000/6 code).
Bold values correspond to the best scoring submission.
Measures: MAP - Mean Average Precision. P - Precision. R - Recall. F1 - F1 score.

Model	Measures							
	With Metastasis				Without Metastasis			
	MAP	P	R	F1	MAP	P	R	F1
Model 2	0.463	0.157	0.549	0.244	0.350	0.119	0.466	0.189
Model 3	0.449	0.159	0.517	0.243	0.333	0.118	0.427	0.184
Model 4	0.455	0.151	0.532	0.235	0.344	0.113	0.445	0.180
Model 5	**0.506**	**0.211**	**0.601**	**0.312**	**0.399**	**0.167**	**0.527**	**0.254**
Model 7	0.459	0.197	0.541	0.289	0.346	0.151	0.456	0.226

4.7 Discussion

Overall, the results achieved in the CANTEMIST-CODING task were better than expected, with the lowest score surpassing to the highest score achieved in my preliminary evaluation. Unfortunately, the competition organizers did not publish a general scoreboard with the results of the remaining competitors, neither have they release a baseline score for each measure, therefore it is impossible to know how well the X-Transformer model worked in comparison with the other participants.

Nevertheless, the results achieved by X-Transformer could be improved. For example, I did not use MER nor any other NER software in the clinical cases in order to extract relevant entities and synonyms from among the corpus. The reason for this was due to the large amount of text that composed each clinical case, thus adding a list of additional words to that text might not improve the results. However, a possible solution could pass by reducing the amount of text from each clinical case before running it through X-Transformer. This could be achieved by using automatic text summarization tools to leave only the essential information about each clinical case. Then, using MER, key entities and related terms could be extracted and include them in the summarized text. This way, by reducing the original clinical case to a smaller and more objective text, along with identified key terms and entities given by the NER tools, it is more likely that the X-Transformer model will be able to achieve better results, like it was shown in Section 3.5.1.2 where the usage of the article's titles in conjunction with MER achieved better results than using the article's abstracts in conjunction with MER.

Chapter 5

Conclusion

Recent advances in deep learning techniques largely improve the performance of text mining solutions developed for NLP tasks across several domains, including the biomedical domain. XMLC is also one of the text mining solutions that was improved with these recent advances in deep learning, leading to state-of-the-art results in several datasets [14, 15]. However, XMLC had not yet been applied to the biomedical domain neither it as been combined with other NLP tools, such as NER software, which lead to the main objective of this thesis: the development of a solution that combines XMLC with NER, and the application to the biomedical and multilingual panorama.

By participating in the BioASQ competitions, I have successfully adapted the state-of-the-art XMLC solution X-BERT [17] along with its most recent version X-Transformer [15] to the biomedical and multilingual biomedical domains. This was achieved through some changes in the algorithm code and by developing a pipeline of scripts to preprocess the data and identify key terms and synonyms from an ontology by using MER [18, 19] before sending the data to the XMLC algorithm, as was detailed in Sections 3.1 and 3.2. However, the results achieved were suboptimal in the most relevant measures evaluated by the competitions. Nonetheless, the XMLC solution achieved high scores in the precision measures in both BioASQ competitions, surpassing most competing systems and even achieving the 1st place in the MiP measure for two consecutive weeks in BioASQ task 8a. This lead to post-competition developments in a way of searching for how the results could be improved in future editions, with an increased focus in the usage of more data to train the models.

After the BioASQ competitions, I had the opportunity to participate in the first edition of the CANTEMIST competition, namely in the CANTEMIST-CODING task which had some similarities with the MESINESP challenge. Therefore, I adapted the solution that was developed for the MESINESP task, as described in Section 4.2, and developed new X-Transformer models trained with different amounts of data and with different pre-trained models, some of them developed for the MESINESP task or using the data given in MESINESP. The results achieved in MAP, the most relevant measure of this challenge, had an high score of 0.506. However, since the competition organizers did not release the results of the remaining competing systems neither a baseline score, it is not possible to fully analyze the performance

of the XMLC solution in the CANTEMIST competition.

With the participation in these international challenges, I was able to combine deep learning XMLC with a NER tool in order to improve the results achieved, and I was also able to apply it to the biomedical and multilingual biomedical panorama, achieving considerable results in the precision measures. However, there is still much room for improvement, namely in the recall measures which were relatively low. Additionally, I was also able to explore the differences in performances between the usage of English and non-English models, with non-English models achieving considerably lower values, which highlights the necessity of non-English datasets, especially in the biomedical sciences.

In the end, this work produced an open-source semantic indexing system based on XMLC that can be used with either English or non-English data, and a total of three publications, namely two workshop publications describing the submitted solutions for the international challenges in which I have participated, and a short paper for the SIIRH 2020 workshop [22], where I proposed a possible application of this pipeline to multilingual biomedical question answering.

5.1 Future Work

As future work, the main goal would be improving the performance of the developed XMLC solution in the biomedical domain. A first step could rely on the usage of more complex MER lexicons. In the BioASQ challenges, the lexicons used were somehow simple, comprising only the MeSH or DeCS ontologies, along with synonyms of the terms and the direct ancestors in the case of DeCS. A new lexicon could comprise not only these elements but also semantic similar terms [86], or even the relations with terms from other ontologies or structured vocabularies with associated DeCS/MeSH terms, in order to retrieve additional information related to the terms and include it in the text. This additional information could be relevant to the XMLC algorithm and consequently lead to a better label classification.

Another possible development would be training X-Transformer models and use them as pre-trained models to develop other X-Transformer models similar to what was done in some models that were submitted in the CANTEMIST- CODING task. As explained in Section 4.4, some of the submitted models were developed using X-Transformer models that were previously developed for the MESINESP task or containing data from the MESINESP task. These models achieved superior results in the preliminary evaluation as seen in Section 4.5, even though the same did not happen in the competition results. Nonetheless, this solution might be able to achieve improved results in challenges similar to MESINESP that require the classification of Spanish biomedical articles.

An additional path to explore is the application of the developed XMLC pipeline to question answering tasks, as proposed by my publication in the SIIRH 2020 workshop [22]. The main hypothesis behind this idea is that, since XMLC algorithms can classify documents with labels related their content, and usually the first step of a question answering system is retrieving documents relevant to the question. By using these labels the question answering system will be able to give more accurate answers, since it had a series of labels to identify the most related documents to the question.

Another further development is combining XMLC with other NLP tools, such as automatic text summarization tools. Through automatic text summarization, it will be possible to reduce the original text to a more concise alternative containing only the key topics of the articles or clinical cases. Combining this text with the terms and entities identified by the NER software and adding them to the processed text, similar to what was done in the BioASQ competitions, it might be possible to further improve the results achieved by X-Transformer.

Finally, a future development that could benefit all of the above proposals is the development of a BERT model for the Spanish biomedical domain by using all of the Spanish biomedical articles given by the BioASQ MESINESP task, which comprises more than 10 million articles. This task would require a great amount of time and hardware processing to be achieved, but it could lead to considerable improvements in the results of not only X-Transformer models, but also other deep learning solutions developed for the Spanish biomedical domain due to a domain-specific language model.

References

[1] Y. Lecun, B. E. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. E. Hubbard, and L. D. Jackel, "Handwritten digit recognition with a back-propagation network," in *Advances in neural information processing systems 2*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1990, pp. 396–404. 1, 11

[2] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, 1998. 1, 11

[3] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, 1986. 1, 10, 11

[4] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, 1997. 1, 11

[5] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," *CoRR*, vol. abs/1810.04805, 2018. [Online]. Available: http://arxiv.org/abs/1810.04805 1, 2, 3, 11, 19, 36, 42

[6] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized BERT pretraining approach," *CoRR*, vol. abs/1907.11692, 2019. [Online]. Available: http://arxiv.org/abs/1907.11692 1, 11, 20

[7] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, and J. Kang, "BioBERT: a pre-trained biomedical language representation model for biomedical text mining," *Bioinformatics*, sep 2019. [Online]. Available: https://academic.oup.com/bioinformatics/advance-article/doi/10.1093/bioinformatics/btz682/5566506 1, 2, 11

[8] I. Beltagy, A. Cohan, and K. Lo, "Scibert: Pretrained contextualized embeddings for scientific text," *CoRR*, vol. abs/1903.10676, 2019. [Online]. Available: http://arxiv.org/abs/1903.10676 1, 2, 11, 21

[9] Z. Yang, Z. Dai, Y. Yang, J. G. Carbonell, R. Salakhutdinov, and Q. V. Le, "Xlnet: Generalized autoregressive pretraining for language understanding," *CoRR*, vol. abs/1906.08237, 2019. [Online]. Available: http://arxiv.org/abs/1906.08237 1, 2, 11, 20

[10] S. Peng, R. You, H. Wang, C. Zhai, H. Mamitsuka, and S. Zhu, "DeepMeSH: deep semantic representation for improving large-scale MeSH indexing," *Bioinformatics*, vol. 32, no. 12, pp. i70–i79, 06 2016. [Online]. Available: https://doi.org/10.1093/bioinformatics/btw294 1

[11] G. Xun, K. Jha, Y. Yuan, Y. Wang, and A. Zhang, "MeSHProbeNet: a self-attentive probe net for MeSH indexing," *Bioinformatics*, vol. 35, no. 19, pp. 3794–3802, 03 2019. [Online]. Available: https://doi.org/10.1093/bioinformatics/btz142 1

[12] F. Soares, M. Villegas, A. Gonzalez-Agirre, M. Krallinger, and J. Armengol-Estapé, "Medical Word Embeddings for Spanish: Development and Evaluation," *Proceedings ofthe 2nd Clinical Natural Language Processing Workshop*, 2019. 2, 13

[13] K. Bhatia, H. Jain, P. Kar, M. Varma, and P. Jain, "Sparse local embeddings for extreme multi-label classification," in *Advances in Neural Information Processing Systems*, 2015. 2, 8, 19, 35

[14] R. You, Z. Zhang, S. Dai, and S. Zhu, "Haxmlnet: Hierarchical attention network for extreme multi-label text classification," *CoRR*, vol. abs/1904.12578, 2019. [Online]. Available: http://arxiv.org/abs/1904.12578 2, 19, 35, 49

[15] W.-C. Chang, H.-F. Yu, K. Zhong, Y. Yang, and I. Dhillon, "Taming Pretrained Transformers for Extreme Multi-label Text Classification," june 2020. [Online]. Available: http://arxiv.org/abs/1905.02331v4 2, 3, 4, 8, 20, 26, 35, 49

[16] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, vol. 2017-Decem, jun 2017, pp. 5999–6009. [Online]. Available: http://arxiv.org/abs/1706.03762 2

[17] W.-C. Chang, H.-F. Yu, K. Zhong, Y. Yang, and I. Dhillon, "X-BERT: eXtreme Multi-label Text Classification with BERT," may 2019. [Online]. Available: http://arxiv.org/abs/1905.02331v2 3, 4, 8, 19, 22, 26, 35, 49

[18] F. M. Couto, L. F. Campos, and A. Lamurias, "MER: a Minimal Named-Entity Recognition Tagger and Annotation Server," *Proceedings of the BioCreative V.5 Challenge Evaluation Workshop*, 2017. 3, 8, 18, 22, 49

[19] F. M. Couto and A. Lamurias, "MER: A shell script and annotation server for minimal named entity recognition and linking," *Journal of Cheminformatics*, 2018. 3, 8, 18, 22, 49

[20] A. Neves, A. Lamurias, and F. M. Couto, "Extreme multi-label classification applied to the biomedical and multilingual panorama," in *Working Notes of CLEF 2020 - Conference and Labs of the Evaluation Forum*, L. Cappellato, C. Eickhoff, N. Ferro, and A. Névéol, Eds. CEUR Workshop Proceedings, 2020. [Online]. Available: http://ceur-ws.org/Vol-2696/paper_67.pdf 4, 5

[21] P. Ruas, A. Neves, V. Andrade, and F. M. Couto, "Lasigebiotm at cantemist: Named entity recognition and normalization of tumour morphology entities and clinical coding of spanish health-related documents," in *Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2020)*, M. Ángel García 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. M. y Gómez, and M. García-Vega, Eds. CEUR Workshop Proceedings, 09 2020. [Online]. Available: http://ceur-ws.org/Vol-2664/cantemist_paper11.pdf 4, 5

[22] A. Neves, A. Lamurias, and F. M. Couto, "Biomedical Question Answering using Extreme Multi-Label Classification and Ontologies in the Multilingual Panorama." in *Proceedings of the Workshop on Semantic Indexing and Information Retrieval for Health from heterogeneous content types and languages co-located with 42nd European Conference on Information Retrieval (ECIR 2020)*, F. M. Couto and M. Krallinger, Eds. Lisbon: CEUR Workshop Proceedings, 2020, pp. 1–3. [Online]. Available: http://ceur-ws.org/Vol-2619/short2.pdf 4, 5, 50

[23] A. Lamurias and F. M. Couto, "Text mining for bioinformatics using biomedical literature," in *Encyclopedia of Bioinformatics and Computational Biology*, S. Ranganathan, M. Gribskov, K. Nakai, and C. Schönbach, Eds. Oxford: Academic Press, 2019, pp. 602 – 611. [Online]. Available: http://www.sciencedirect.com/science/article/pii/B9780128096338204093 7

[24] A.-H. Tan, "Text Mining: The state of the art and the challenges," *Proceedings of the PAKDD 1999 Workshop on Knowledge Disocovery from Advanced Databases*, 1999. 7

[25] A. Hotho, A. Nürnberger, and G. Paaß, "A Brief Survey of Text Mining," *LDV Forum - GLDV Journal for Computational Linguistics and Language Technology*, 2005. 7

[26] S. Harabagiu and A. Hickl, "Methods for using textual entailment in open-domain question answering," in *COLING/ACL 2006 - 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference*, 2006. 7

[27] A. Celikyilmaz, M. Thint, and Z. Huang, "A graph-based semi-supervised learning for question-answering," in *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*. Suntec, Singapore: Association for Computational Linguistics, Aug. 2009, pp. 719–727. [Online]. Available: https://www.aclweb.org/anthology/P09-1081 7

[28] J. Li, A. Sun, J. Han, and C. Li, "A Survey on Deep Learning for Named Entity Recognition," *IEEE Transactions on Knowledge and Data Engineering*, 2020. 7, 18

[29] P. Corbett and J. Boyle, "Chemlistem: Chemical named entity recognition using recurrent neural networks," *Journal of Cheminformatics*, 2018. 8, 19

[30] M. Krallinger, M. Pérez-Pérez, G. Pérez-Rodríguez, A. Blanco-Míguez, F. Fdez-Riverola, S. Capella-Gutiérrez, A. Lourenço, and A. Valencia, "The biocreative v.5 evaluation workshop: tasks, organization, sessions and topics," in *Proceedings of the BioCreative V.5 challenge evaluation workshop*, 2017, pp. 8–10. 8, 19

[31] M. Fresko, B. Rosenfeld, and R. Feldman, "A hybrid approach to NER by MEMM and manual rules," in *International Conference on Information and Knowledge Management, Proceedings*, 2005. 8

[32] M. Habibi, L. Weber, M. Neves, D. L. Wiegandt, and U. Leser, "Deep learning with word embeddings improves biomedical named entity recognition," in *Bioinformatics*, 2017. 8

[33] G. Tsoumakas and I. Katakis, "Multi-label classification: An overview," in *International Journal of Data Warehousing and Mining*, 2007. 8

[34] A. Clare and R. D. King, "Knowledge discovery in multi-label phenotype data," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2001. 8

[35] G. Tsoumakas, I. Katakis, and I. Vlahavas, "Effective and efficient multilabel classification in domains with large number of labels," in *Proc. ECML/PKDD 2008 Workshop on Mining Multidimensional Data (MMD'08)*, 2008. 8

[36] S. Kannan, V. Gurusamy, S. Vijayarani, J. Ilamathi, M. Nithya, S. Kannan, and V. Gurusamy, "Preprocessing Techniques for Text Mining," *International Journal of Computer Science & Communication Networks*, 2015. 9

[37] T. Oshikiri, "Segmentation-free word embedding for unsegmented languages," in *EMNLP 2017 - Conference on Empirical Methods in Natural Language Processing, Proceedings*, 2017. 9

[38] B. Xie, Q. Ding, H. Han, and D. Wu, "MiRCancer: A microRNA-cancer association database constructed by text mining on literature," *Bioinformatics*, 2013. 9

[39] A. Vlachos, "Tackling the biocreative2 gene mention task with conditional random fields and syntactic parsing," *Proceedings of the Second BioCreative Challenge Evaluation Workshop; 23 to 25 April 2007*, 01 2007. 9

[40] A. Jivani, "A comparative study of stemming algorithms," *International Journal of Computer Technology and Applications*, vol. 2, pp. 1930–1938, 11 2011. 9

[41] M. F. Porter, "An algorithm for suffix stripping," *Program*, vol. 14, pp. 130–137, 1980. 9

[42] P. Majumder, M. Mitra, S. K. Parui, G. Kole, P. Mitra, and K. Datta, "Yass: Yet another suffix stripper," *ACM Trans. Inf. Syst.*, vol. 25, no. 4, p. 18–es, Oct. 2007. [Online]. Available: https://doi.org/10.1145/1281485.1281489 9

[43] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 3rd ed. USA: Prentice Hall Press, 2009. 10

[44] D. P. Kingma, D. J. Rezende, S. Mohamed, and M. Welling, "Semi-supervised learning with deep generative models," in *Advances in Neural Information Processing Systems*, 2014. 10

[45] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *The Bulletin of Mathematical Biophysics*, 1943. 10

[46] D. E. Rumelhart, B. Widrow, and M. A. Lehr, "The Basic Ideas in Neural Networks," *Communications of the ACM*, 1994. 10

[47] D. Graupe, *Principles of Artificial Neural Networks*, 3rd ed. WORLD SCIENTIFIC, 2013. [Online]. Available: https://www.worldscientific.com/doi/abs/10.1142/8868 10

[48] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature Cell Biology*, vol. 521, no. 7553, pp. 436–444, May 2015. 11

[49] G. Mesnil, Y. Dauphin, K. Yao, Y. Bengio, L. Deng, D. Hakkani-Tur, X. He, L. Heck, G. Tur, D. Yu, and G. Zweig, "Using recurrent neural networks for slot filling in spoken language understanding," *IEEE Transactions on Audio, Speech and Language Processing*, 2015. 11

[50] T. Young, D. Hazarika, S. Poria, and E. Cambria, "Recent trends in deep learning based natural language processing," *CoRR*, vol. abs/1708.02709, 2017. [Online]. Available: http://arxiv.org/abs/1708.02709 11

[51] K. W. CHURCH, "Word2vec," *Natural Language Engineering*, vol. 23, no. 1, p. 155–162, 2017. 11

[52] J. Pennington, R. Socher, and C. D. Manning, "GloVe: Global vectors for word representation," in *EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, 2014. 11

[53] M. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, Jun. 2018, pp. 2227–2237. [Online]. Available: https://www.aclweb.org/anthology/N18-1202 11, 19, 20

[54] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, and J. Brew, "Huggingface's transformers: State-of-the-art natural language processing," *ArXiv*, vol. abs/1910.03771, 2019. 12, 18

[55] N. Baumann, "How to use the medical subject headings (mesh)," *International Journal of Clinical Practice*, vol. 70, no. 2, pp. 171–174, 2016. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/ijcp.12767 12

[56] A. Conneau, R. Rinott, G. Lample, A. Williams, S. Bowman, H. Schwenk, and V. Stoyanov, "XNLI: Evaluating cross-lingual sentence representations," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, Oct.-Nov. 2018, pp. 2475–2485. [Online]. Available: https://www.aclweb.org/anthology/D18-1269 13

[57] J. Veiga de Cabo, "El indice bibliográfico español de ciencias de la salud. Cooperación con Latinoamérica," *Revista Española de Salud Pública*, vol. 73, pp. 529 – 532, 09 1999. [Online]. Available: http://scielo.isciii.es/scielo.php?script=sci_arttext&pid=S1135-57271999000500001&nrm=iso 13, 25

[58] W. H. Organization, *International classification of diseases for oncology (ICD-O)*, 3rd ed. World Health Organization, 2013, 1st revision. 14

[59] C. X. Ling and V. S. Sheng, *Class Imbalance Problem*. Boston, MA: Springer US, 2010, pp. 171–171. [Online]. Available: https://doi.org/10.1007/978-0-387-30164-8_110 15

[60] F. Brucker, F. Benites, and E. Sapozhnikova, "An empirical comparison of flat and hierarchical performance measures for multi-label classification with hierarchy extraction," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2011. 16

[61] V. V. Asch, "Macro-and micro-averaged evaluation measures," *Belgium: CLiPS*, 2013. 16

[62] A. Kosmopoulos, I. Partalas, É. Gaussier, G. Paliouras, and I. Androutsopoulos, "Evaluation measures for hierarchical classification: a unified view and novel approaches," *CoRR*, vol. abs/1306.6802, 2013. [Online]. Available: http://arxiv.org/abs/1306.6802 17

[63] P. G. Ipeirotis, L. Gravano, and M. Sahami, "Probe, count, and classify: Categorizing hidden web databases," in *Proceedings of the 2001 ACM SIGMOD International Conference on Management of Data*, ser. SIGMOD '01. New York, NY, USA: Association for Computing Machinery, 2001, p. 67–78. [Online]. Available: https://doi.org/10.1145/375663.375671 17

[64] S. Kiritchenko, S. Matwin, and A. F. Famili, "Functional annotation of genes using hierarchical text categorization," in *in Proc. of the BioLINK SIG: Linking Literature, Information and Knowledge for Biology (held at ISMB-05*, 2005. 17

[65] N. Cesa-Bianchi, C. Gentile, A. Tironi, and L. Zaniboni, "Incremental algorithms for hierarchical classification," in *Proceedings of the 17th International Conference on Neural Information Processing Systems*, ser. NIPS'04. Cambridge, MA, USA: MIT Press, 2004, p. 233–240. 17

[66] A. V. Aho, J. E. Hopcroft, and J. D. Ullman, "On finding lowest common ancestors in trees," in *Proceedings of the Fifth Annual ACM Symposium on Theory of Computing*, ser. STOC '73. New York, NY, USA: Association for Computing Machinery, 1973, p. 253–265. [Online]. Available: https://doi.org/10.1145/800125.804056 17

[67] A. Turpin and F. Scholer, "User performance versus precision measures for simple search tasks," in *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '06. New York, NY, USA: Association for Computing Machinery, 2006, p. 11–18. [Online]. Available: https://doi.org/10.1145/1148170.1148176 17

[68] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016. 18

[69] D. Amodei, S. Ananthanarayanan, R. Anubhai, J. Bai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, Q. Cheng, G. Chen, J. Chen, J. Chen, Z. Chen, M. Chrzanowski, A. Coates, G. Diamos, K. Ding, N. Du, E. Elsen, J. Engel, W. Fang, L. Fan, C. Fougner, L. Gao, C. Gong, A. Hannun, T. Han, L. V. Johannes, B. Jiang, C. Ju, B. Jun, P. LeGresley, L. Lin, J. Liu, Y. Liu, W. Li, X. Li, D. Ma, S. Narang, A. Ng, S. Ozair, Y. Peng, R. Prenger, S. Qian, Z. Quan, J. Raiman, V. Rao, S. Satheesh, D. Seetapun, S. Sengupta, K. Srinet, A. Sriram, H. Tang, L. Tang, C. Wang, J. Wang, K. Wang, Y. Wang, Z. Wang, Z. Wang, S. Wu, L. Wei, B. Xiao, W. Xie, Y. Xie, D. Yogatama, B. Yuan, J. Zhan, and Z. Zhu, "Deep speech 2: End-to-end speech recognition in english and mandarin," in *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ser. ICML'16. JMLR.org, 2016, p. 173–182. 18

[70] M. K. Leung, H. Y. Xiong, L. J. Lee, and B. J. Frey, "Deep learning of the tissue-regulated splicing code," *Bioinformatics*, 2014. 18

[71] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," *Journal of Machine Learning Research*, 2011. 18

[72] P. L. Whetzel, N. F. Noy, N. H. Shah, P. R. Alexander, C. Nyulas, T. Tudorache, and M. A. Musen, "BioPortal: Enhanced functionality via new Web services from the National Center for Biomedical Ontology to access and use ontologies in software applications," *Nucleic Acids Research*, 2011. 18

[73] A. P. Quimbaya, A. S. Múnera, R. A. G. Rivera, J. C. D. Rodríguez, O. M. M. Velandia, A. A. G. Peña, and C. Labbé, "Named Entity Recognition over Electronic Health Records Through a Combined Dictionary-based Approach," in *Procedia Computer Science*, 2016. 18

[74] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer, "Neural architectures for named entity recognition," in *2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL HLT 2016 - Proceedings of the Conference*, 2016. 18

[75] J. Fisher and A. Vlachos, "Merge and label: A novel neural network architecture for nested NER," in *ACL 2019 - 57th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference*, 2020. 19

[76] C. Walker, S. Strassel, J. Medero, and K. Maeda, "Ace 2005 multilingual training corpus ldc2006t06," Web Download. Philadelphia: Linguistic Data Consortium, 2006. 19

[77] W. Ralph, M. Palmer, M. Mitchell, E. Hovy, S. Pradhan, L. Ramshaw, X. Nianwen, T. Ann, J. Kaufman, M. Franchini, M. El-Bachouti, R. Belvin, and A. Houston, "Ontonotes release 5.0 ldc2013t19," Web Download. Philadelphia: Linguistic Data Consortium, 2013. 19

[78] X. Li, J. Feng, Y. Meng, Q. Han, F. Wu, and J. Li, "A unified mrc framework for named entity recognition," 2019. 19

[79] Y. Prabhu, A. Kag, S. Harsola, R. Agrawal, and M. Varma, "Parabel: Partitioned label trees for extreme classification with application to dynamic search advertising," in *2018 World Wide Web Conference*, International World Wide Web Conferences Steering Committee. International World Wide Web Conferences Steering Committee, April 2018, pp. 993–1002. [Online]. Available: https://www.microsoft.com/en-us/research/publication/parabel-partitioned-label-trees-for-extreme-classification-with-application-to-dynamic-search-advertising/ 19, 20, 35

[80] J. Liu, W. C. Chang, Y. Wu, and Y. Yang, "Deep learning for extreme multi-label text classification," in *SIGIR 2017 - Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2017. 19, 35

[81] Y. Kim, "Convolutional neural networks for sentence classification," in *EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, 2014. 19, 35

[82] R. You, S. Dai, Z. Zhang, H. Mamitsuka, and S. Zhu, "Attentionxml: Extreme multi-label text classification with multi-label attention based recurrent neural networks," *CoRR*, vol. abs/1811.01727, 2018. [Online]. Available: http://arxiv.org/abs/1811.01727 19, 35

[83] D. Martinez and Y. Li, "Information extraction from pathology reports in a hospital setting," in *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, ser. CIKM '11. New York, NY, USA: Association for Computing Machinery, 2011, p. 1877–1882. [Online]. Available: https://doi.org/10.1145/2063576.2063846 41

[84] S. Sánchez Seda, F. d. P. Pérez León, J. Moreno Conde, M. C. Gutiérrez Ruiz, J. Martín Sánchez, G. Rodríguez, J. A. Pérez Simón, and C. L. Parra Calderón, "Plataforma para la extracción automática y codificación de conceptos dentro del ámbito de la oncohematología (proyecto coco)," 2018-09. 41

[85] J. Cañete, G. Chaperon, R. Fuentes, and J. Pérez, "Spanish pre-trained bert model and evaluation data," in *to appear in PML4DC at ICLR 2020*, 2020. 42

[86] F. M. Couto and A. Lamurias, "Semantic similarity definition," in *Encyclopedia of Bioinformatics and Computational Biology*, S. Ranganathan, M. Gribskov, K. Nakai, and C. Schönbach, Eds. Oxford: Academic Press, 2019, pp. 870 – 876. [Online]. Available: http://www.sciencedirect.com/science/article/pii/B9780128096338204019 50