



Universidad Politécnica
de Madrid

**Escuela Técnica Superior de
Ingenieros Informáticos**



Máster Universitario en Inteligencia Artificial

Trabajo Fin de Máster

**Named Entity Recognition and
Normalization in Biomedical Literature: A
Practical Case in SARS-CoV-2 Literature**

Autor(a): Álvaro Alonso Casero
Tutor(a): Óscar Corcho
Tutor(a): Carlos Badenes-Olmedo

Madrid, «julio 2021»

Este Trabajo Fin de Máster se ha depositado en la ETSI Informáticos de la Universidad Politécnica de Madrid para su defensa.

Trabajo Fin de Máster

Máster Universitario en Inteligencia Artificial

Título: Named Entity Recognition and Normalization in Biomedical Literature: A Practical Case in SARS-CoV-2 Literature

«julio 2021»

Autor(a): Álvaro Alonso Casero

Tutor(a): Óscar Corcho

Tutor(a): Carlos Badenes-Olmedo

OEG

ETSI Informáticos

Universidad Politécnica de Madrid

Agradecimientos

Este trabajo está apoyado por el proyecto Drugs4Covid++ (<https://drugs4covid.oeg.fi.upm.es>), financiado por las ayudas Fundación BBVA a equipos de investigación científica SARS-CoV-2 y COVID-19.

Resumen

Las tareas de recuperación de información se han convertido en una herramienta esencial para la investigación biomédica. La tendencia creciente en el número de publicaciones ha hecho necesario desarrollar e implementar estas herramientas para ayudar a los investigadores a mantenerse al día con los últimos avances en su campo. Una de las tareas de minería de texto más fundamentales en la recuperación de información en el área biomédica es el reconocimiento de entidades nombradas biomédicas, como enfermedades, sustancias químicas o genes, lo que se conoce comúnmente como BioNER. Como complemento a este reconocimiento, las entidades detectadas suelen vincularse a bases de datos curadas en un proceso denominado linkeo o normalización de entidades (BioNEN). Las metodologías aplicadas en BioNER han ido evolucionando durante años hasta llegar al estado del arte actual, que se basa fundamentalmente en el uso de modelos de lenguaje como BERT que son preentrenados en el campo biomédico para especificar su conocimiento subyacente. La revisión realizada recorrerá cómo se realizan las tareas de NER y como ha sido su evolución hasta llegar a estos modelos del estado del arte actual. Esta revisión nos ha ayudado a implementar un sistema que se basa en el actual modelo del estado del arte, BioBERT. Un modelo BioBERT se ha puesto a punto para realizar la tarea NER para cada una de las clases de entidades consideradas: enfermedades, productos químicos y genética. Los resultados han sido normalizados mediante una búsqueda de índice inverso en una base de datos construida con la unión y mapeo de términos de múltiples fuentes. Este sistema se aplica en dos casos prácticos, uno como pieza central en una plataforma web a la que se pueden enviar textos para ser procesados por el sistema y otro para procesar el corpus CORD-19, compuesto por artículos relacionados con el SARS-CoV-2. Se ha evaluado el sistema, mostrando una puntuación F1 de 0,86 en PGxCorpus (en Micro-Average para coincidencias parciales, los resultados variarán ligeramente dependiendo del escenario considerado). Con un análisis de errores, concluimos que la mayoría de los errores se observaron debido a la detección incorrecta de los límites de las entidades.

Disponibilidad e Implementación

El código fuente se puede encontrar en: <https://github.com/librairy/bio-ner>.

La plataforma web está disponible en: <https://librairy.github.io/bio-ner/>.

El corpus CORD-19 procesado está disponible en: <http://librairy.linkeddata.es/solr/#/cord19-paragraphs/core-overview>

Los modelos usados en el sistema se encuentran en repositorios en Huggingface:

-
- **Enfermedades:** https://huggingface.co/alvaroalon2/biobert_diseases_ner
 - **Químicos:** https://huggingface.co/alvaroalon2/biobert_chemical_ner
 - **Genético:** https://huggingface.co/alvaroalon2/biobert_genetic_ner

Abstract

Information retrieval tasks have become an essential tool for biomedical research. The growing tendency in the number of publications has made it necessary to develop and implement these tools to help researchers to keep up with the latest advances in their field. One of the most fundamental text-mining tasks in information retrieval in the biomedical area is the recognition of biomedical named entities like diseases, chemicals, genes... which is commonly known as BioNER. Complementary to this recognition, detected entities are usually linked to curated databases in a process called entity linking or normalization (BioNEN). Methodologies applied in BioNER have been evolving for years until the current state-of-the-art, which is mainly based on the use of language models such as BERT that are pretrained in the biomedical field to specify its underlying knowledge. A review will walk through how the NER tasks are carried out and about its evolution until these current state-of-the-art models. This review has allowed us to implement a system which is based on the current state-of-the-art model, BioBERT. One BioBERT model has been fine-tuned to perform NER task for each of the considered entity classes: diseases, chemicals and genetics. Results have been normalized through an inverse index search in a built database in which we join and map terms from multiple sources. This system is applied in two practical cases, a first one as the core piece in a web platform where text can be sent to be processed by the system and a second one for processing the CORD-19 corpus, composed by papers related to SARS-CoV-2. The system has been evaluated, showing an F1-Score of 0,86 in PGxCorpus (in Micro-Average for partial matches, the results will slightly vary depending on the considered scenario). With an error analysis, we conclude that most errors were observed to be due to incorrect boundary detection.

Availability and Implementation

Source code can be found on: <https://github.com/librairy/bio-ner>.

Web Platform is available at: <https://librairy.github.io/bio-ner/>.

CORD-19 processed corpus is available at: <http://librairy.linkeddata.es/solr/#/cord19-paragraphs/core-overview>

Fine-tuned models used in the system can be found on Huggingface repositories:

- Diseases: https://huggingface.co/alvaroalon2/biobert_diseases_ner
- Chemicals: https://huggingface.co/alvaroalon2/biobert_chemical_ner
- Genetics: https://huggingface.co/alvaroalon2/biobert_genetic_ner

Contents

1	Introduction	1
1.1	Objectives	3
2	State of the Art of Biomedical NERs	5
2.1	Introduction	5
2.2	Methods	6
2.2.1	Methodology	7
2.2.2	Previous Work	8
2.2.3	Data pipeline	9
2.2.3.1	Pre-processing	9
2.2.3.2	Feature Processing	11
2.2.3.3	BioNER Models	20
2.2.3.4	Post-processing	35
2.2.4	Evolution of methodologies	36
2.3	Datasets	36
2.3.1	Entities	38
2.3.2	Performance	40
2.4	Source Code Availability	44
3	Biomedical Named Entity Recognition and Normalization System Implementation	47
3.1	State-of-the-art method reutilization	47
3.2	Development	48
3.2.1	Resources	49
3.2.1.1	Previous work reusability	49
3.2.1.2	Libraries and Services	49
3.2.2	Workflow	50
3.3	Creation of BioNER models	51
3.3.1	Language representation	51
3.3.2	Fine-Tuning	57
3.3.2.1	Selected corpus	57
3.4	Implementation of BioNER/BioNEN system	59
3.4.1	Pre-processing	59
3.4.2	BioNER Modelling	60
3.4.3	Post-processing	61
3.4.3.1	Boundaries correction and Overlapping resolution	61
3.4.3.2	Capturing new entities with Regular Expressions	61
3.4.4	Normalization	62

3.5	Deployment as Web Platform	66
3.5.1	AJAX calls	66
3.6	CORD-19 Annotation	66
4	Evaluation and Conclusions	71
4.1	NER Evaluation	71
4.1.1	Datasets	71
4.1.1.1	PGxCorpus	71
4.1.1.2	COVID-19 MLIA @ Eval	72
4.1.2	Evaluation Metrics	73
4.1.2.1	Exact-match Evaluation	74
4.1.2.2	Partial-match Evaluation	74
4.1.3	Experimental Results	74
4.1.4	Discussion	75
4.1.4.1	PGxCorpus	75
4.1.4.2	COVID-19 MLIA	76
4.2	Normalization Results	77
4.3	CORD-19 Corpus Annotation Results	78
4.4	Future Work	81
4.5	Conclusions	82
	Bibliography	82
	Appendix	95
.1	SOTA retrieved publications	95

Chapter 1

Introduction

Since most of the biological discoveries are disseminated through academic papers, every researcher in the life sciences must be able to locate and interpret important information in the literature. The number of biomedical articles available in the Internet has been increasing year by year. Thousands of new biomedical articles appear every day, as it can just be seen in Fig. 1.1 in the case of PubMed Central (PMC¹), which is one of the largest archives in the biomedical literature. Besides that, the appearance of the SARS-CoV-2 virus and the COVID-19 pandemic has not done more than increase this growing tendency since experts from throughout the world have put their efforts to find solutions to the expansion and development of this infectious virus and its associated consequences. Therefore, this has supposed a boost in the amount of biomedical literature in little time, overall related to the COVID-related problems in different areas.

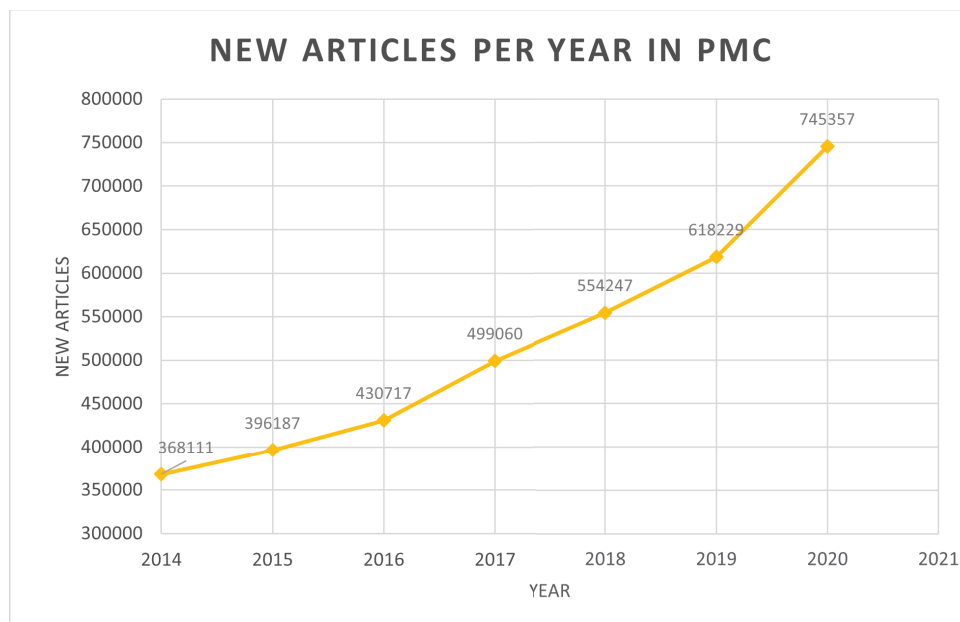


Figure 1.1: Growing tendency of the number of new articles in PMC since 2014. Data taken from PMC

¹<https://pubmed.ncbi.nlm.nih.gov/>

The large amount of data underlying the biomedical literature needs to be adequately managed to ease its usage to healthcare professionals. This leads us to the requirement of processing this data to turn it into useful knowledge. The development of Natural Language Processing (NLP) has helped and given us different ways of approaching to this data processing, facilitating research productivity through the extraction of the key information underlying texts and turning it into structured knowledge that can be understood by humans.

One of the main approaches applied in NLP is the recognition of relevant entities found in the literature, which is commonly known as Named Entity Recognition (NER). Meaningful terms in a domain are called named entities. NER is a task which has as objective automatically identifying these named entities in a text and classifying them into predefined entity classes. In the biomedical domain, the entities can be genes, drugs, diseases, etc... and the task is more specifically known as BioNER. Once the text has been processed and the entities recognized, there are several applications of the processed data helping in the application of subsequent NLP approaches. Since Biomedical terms are referred to specific kinds of drugs, gene mutations, medical concepts, etc... it is usual to classify these terms, in order to avoid ambiguities, following a compendium of controlled and curated biomedical vocabularies such as MESH² or another more specific such as ATC³ in the case of chemicals. This task is commonly known as Entity Linking or Named Entity Normalization (NEN) and, if it is applied in the biomedical domain, BioNEN. Its objective will be to obtain a common framework for recognized entities to avoid all kinds of possible ambiguities and extend the available information regarding each of the retrieved entities. Therefore, once these approaches have been applied to the biomedical literature, a set of normalized concepts related to the biomedical field could be applied to take advantage of Information Retrieval processes like the creation of efficient search algorithms, content classification, Knowledge-Graph construction or processes related to the Semantic Web like the discovery of new knowledge, mixed integration of data sources, etc... among a long list of applications [57].

The advances in computational power and the presence of increasingly more data have boosted the development of data-centric fields such as Machine Learning. These advances have not only boosted the results obtained in a wide range of fields but also have facilitated the automation of carrying out these tasks, thanks to the use of large amounts of data in the training of these algorithms which is the step where they learn to perform the task. Machine Learning field has also experimented great advances in recent years in text processing tasks, evolving the way Natural Language Processing is raised. For those purposes, generally the more data we have, the best intelligent-based systems will appear to overcome the difficulties engaged in the understanding of language. Moreover, as aforementioned, the extending of more computational capable machines and cloud computation has not done more than increase the usage of this large amount of data for the continuous improvement of ML data-based approaches. All of it has taken us to experiment great advances in areas such as NLP, with the appearance of each time more advanced techniques with a deeper understanding of the language allowing us to design more expert systems that require this language understanding and the knowledge underlying it.

All these advances have been firstly studied throughout a state-of-the-art review in

²<https://www.nlm.nih.gov/mesh>

³<https://www.whocc.no/>

Named Entity Recognition of Biomedical terms (BioNER). Since NER is one of the most widely used NLP tasks, more recent advances in text processing generally appear along the techniques used in NER. Following this review, an implementation of a BioNER/BioNEN system has been proposed jointly with a pair of practical cases where this system is used and tested: a Web Platform to facilitate its usage and a CORD-19 annotator where the system is used for annotating and normalizing the biomedical terms found on a CORD-19 corpus [135].

1.1 Objectives

As a means of clarifying the goals of the project development, a set of objectives have been proposed aiming to meet them along the project:

- Show the importance of NER as an starting point in subsequent processes.
- Study why NER must be specified in fields like the biomedical.
- Expose an state-of-the-art review in which is shown how advances in BioNER area have evolved until nowadays state-of-the-art models.
- Illustrate how NER task is carried out and how the evolution of this process has been.
- Review each of the retrieved methodologies stating the advances exposed in its publications.
- Compare the different methodologies retrieved in the review and conclude with the state-of-the-art technique.
- Show benefits underlying the state-of-the-art technique which differentiate it from previous techniques.
- Expose the way the retrieved techniques are implemented and study its availability for later use.
- Create BioNER models based on the state-of-the-art techniques retrieved in the review.
- Implement a system for multiclass entity recognition and normalization in biomedical domain.
- Optimize the creation workflow in such a way the system performance is maximized. Adapt BioNER selected model within a system pipeline. Design post-processing rules to enhance the results. Build a collection of terms for each selected entity class to use them in a database for later normalization. Implement a retrieval search engine for the normalization with these retrieved terms.
- Make use of this system in two practical cases:
 - As part of a web platform in which its use is facilitated, requiring the user just to send the text which it is wanted to be annotated and normalized.
 - In the annotation and normalization of the CORD-19 [135] corpus which is composed of SARS-CoV-2 related articles.

- Analyze the results and conclusions obtained in the implementation of this system in the use cases.
- Present a set of future directions in which the project could continue.

Chapter 2

State of the Art of Biomedical NERs

2.1 Introduction

Requirements in the biomedical field about structuring in domain knowledge have increased along with the growing amount of literature available. These requirements are related to the difficulties that experts in some of the biomedical-related areas have found to keep up with the advances of its research areas. Therefore, a suitable management of large volumes of data and the structuring of its underlying knowledge has become an essential process that has taken to focus the development of NLP tools.

Therefore, in this context, Natural Language Processing solutions arise with the aim of making an efficient and structured use of the large amount of knowledge underlying the biomedical literature, making it easier for researchers and health-care professionals to carry out their work. BioNER is one of the most widely used solutions for these purposes since lots of subsequent techniques are based on this. BioNER implies an automatic search of biomedical terms to classify them into entities of interest such as *Genes*, *Chemicals*, *Diseases*, *Species*, etc. . . This term identification is usually followed by a BioNEN process for the normalization of these entities into a unified term usually held in a curated database. These processes play a key role in subsequent tasks such as Information Extraction, Question-Answering, Information Retrieval, Relation Extraction, Knowledgebase population, Semantic search. . . [57] which are tasks that usually take advantage of the results obtained in BioNER and BioNEN tasks. Consequently, it will be vital for the correct development of subsequent actions, the implementation of accurate BioNER/BioNEN models.

The intrinsic characteristics of the biomedical corpus, because of its highly specialized domain linguistical characteristics, cause the presence of challenges on text processing tasks. BioNER task is highly affected by these challenges since biomedical terms are usually very specific and are just found on the biomedical domain and therefore its peculiarities are just found on in-domain literature. These difficulties underlying the biomedical domain were collected in Zhou et al. [149] and can be summarized as follows:

- *Highly specialized terms*: Since the biomedical domain it is a highly specialized domain, most of the terms are exclusive of these kinds of texts, making it

difficult that general domain knowledge could be used to properly identify and classify specific domain concepts.

- *Sharing of nouns*: Some conjunction and disjunction sentences present a common head noun for several words, which in the case of biomedical entities could hinder its identification in separate entities. For instance, "5kb and 17kb viruses" refers to "5kb viruses and 17kb viruses".
- *Non-standardized naming convention*: the reference to some entities can be done in a descriptive way using various words to describe a specific entity. Moreover, the way you describe an entity or the spelling form used makes that the number of ways to refer to the same entity may be high. In some domains, it is usually recommended to use standardized terms to avoid these ambiguities, but this kind of practice it is not always found in biomedical corpora. Therefore, this makes it difficult to identify some boundaries in the entities and the correct identification of them with the correct tag. For example, "*N - acetyl - β - D - glucosamine*", "*N - Acetylglucosamine*", "*C₁₈H₁₅NO₆*", " *β - D - (Acetylamino) - 2 - deoxy - glucopyranose*" and "*Amide derivative of the monosaccharide glucose*" refers to the same concept which could be identified in PubChem¹ Database as ID: 24139².
- *Abbreviations*: In the Biomedical Domain it is a common practice the use of abbreviations to refer to some domain concepts. The drawback is that not always these abbreviations are enough to identify the correct term due to some ambiguities or irregular abbreviations in its usage making it necessary to give them meaning based on the context they appear. For instance, "CVA" could be referred to "*cardiovascular accident*", "*cerebrovascular accident*" or "*costovertebral angle*".

From these difficulties rise the need for adapting general corpora NER tools to the biomedical domain since this adaptation will allow higher performance because of its high specificity to the peculiarities of the biomedical domain. Moreover, BioNER implementations in these processing pipelines are essential since with this technique we aim to normalize the appearance of polysemy in text formerly exposed, referring to a curated term for the different ways some concepts may appear in the text.

In this literature review, we want to discover the methods taken into account for designing Natural Language Processing pipelines related to entity recognition of biomedical terms within biomedical articles. We have also looked for proper corpus on which these methods could be trained and evaluated and the most widely recognized entities within these corpus have also been set. Finally, it was established how is the availability and license of the retrieved methods to use them within an NLP system if desired.

2.2 Methods

In this review, the focus will be on reviewing the most relevant publications within the BioNER field to establish the state-of-the-art methodologies in this field in the biomedical area. The research question that has been established to be answered is:

¹<https://pubchem.ncbi.nlm.nih.gov/>

²<https://pubchem.ncbi.nlm.nih.gov/compound/24139>

What is the current scenario for identifying named entities in the biomedical domain? As a means of clarifying the objectives taken into account in the development of this section, some supplementary questions have been added:

Question 1: What are the challenges in identifying named entities in the biomedical domain?

Question 2: What is the state-of-the-art technique?

From both questions, we aim to extract what is the main method in BioNER from which the following question has to be answered:

Question 3: What are its characteristics and results and how does it improve and differs from previous methods?

2.2.1 Methodology

Some criteria have to be set to find and restrict the publications and their related work taken in mind. The source used to carry out the searching of articles has been Google Scholar since its results contain different journal sources such as OUP³, BMC⁴, Elsevier⁵, etc. . . and publications in different conferences. Both inclusion and exclusion criteria are the key points on which our methodology for this review is based on.

- **Inclusion criteria:** to consider an article as a candidate in the review, it has to meet the following inclusion criteria (IC):
 - IC1: Be searchable via Google Scholar⁶.
 - IC2: The articles taken are the ones that appeared after searching in Google Scholar "Named Entity Recognition" AND "Biomedical". These terms are searched along the entire publication: title, abstract and body. The articles taken are the 300-top results that appeared.
- **Exclusion Criteria:** As a means of refining the search, some exclusion criteria (EC) were set and therefore the retrieved articles which do not meet these criteria are excluded from the review:
 - EC1: Any kind of document which is not a peer-reviewed scientific article.
 - EC2: Articles before 2011 were not considered since they were widely covered in several previous surveys [35] [56] and they are no more in the state-of-the-art trends. Furthermore, publications after January 2021 were not considered as they concur with the writing of this review.
 - EC3: It has to be relevant in terms of the number of citations, being that set to 30 citations in Google Scholar to consider it. Since more recent papers (mid-2020) are difficult to meet these criteria because of the temporal proximity, more lax criteria have been adopted establishing this threshold to 15 citations in that case.

³<https://academic.oup.com/journals>

⁴<https://www.biomedcentral.com/journals>

⁵<https://www.sciencedirect.com/browse/journals-and-books>

⁶<https://scholar.google.com/>

- *EC4*: Articles that are not published in $\geq B$ rank Conferences (by <http://portal.core.edu.au/conf-ranks/> Conference Rank) or Q1 rank Journals (by <https://www.scimagojr.com/> Journal Rank) are filtered out.
 - *EC5*: Articles which after a manual inspection in the following manner did not result to be relevant are not considered: Publications titles were read, if it turns out that it is not relevant the abstract is read in order to make sure of its relevance. If neither title nor abstract result to be relevant then this article is finally excluded. This relevancy was established attending to the fact of presence of the following keywords in the abstract or title: ("Named Entity Recognition" OR "NER" OR "recognition of [ENTITY]"⁷) AND ("method" OR "methodology" OR "technique").
- **Steps in criteria application**: These criteria are applied following these successive steps:
 1. We first apply IC1, IC2 and EC2 obtaining 300 results. These are the 300 more relevant results in Google Scholar for our query in the timestamp established.
 2. Afterwards, we apply criteria EC1, EC3, EC4 and EC5 filtering out from the previous 300 results, the ones that do not meet these excluding criteria.

Results: In the case of applying the previous criteria in the described manner to the search of the BioNER literature, we obtained 300 articles in the first step which were filtered out in step 2 resulting on 27 articles (see Appendix .1).

2.2.2 Previous Work

Most efforts in previous related surveys are widely focused on general domain adapted tools. Nadeau and Sekine [100] reviews the historical evolution of NER since it was first coined at MUC-6 [60] and how it has evolved until the year 2007, progressing from hand-crafted approaches as rules to more automated methods such as Machine Learning. Besides that, it performs an analysis of the different text features often used to represent the different tokens in a text. In Mohit [99] it is confirmed the trend of statistical methods of Machine Learning models as the most promising approach in NER systems. The author also states, within this tendency, the rise of probabilistic methods like the Hidden Markov Models and its limitations in the usage of contextual information and the emergence of methods that improve some limitations like the Conditional Random Fields (CRFs) models. In Sharnagat [122], it is introduced the appearance of distributed word representations for an automated word feature building based on different usages of Neural Networks. In Goyal et al. [57], an extensive review in several domains and languages is done showing that in recent approaches, until 2016, supervised models have definitively surpassed hand-crafted rules models. It can be observed that the clear trend throughout domains and languages is the usage of CRFs-based models since they capture conditioned dependencies in text giving some importance to the contextual information along with the text. Nevertheless, the recent advances in Deep Learning methods are not included. In Yadav and Bethard [143], feature-inferring neural network systems are compared with earlier known methods. They conclude that in hardly any classic feature-engineered model

⁷Name of biomedical entities such as chemicals, diseases, genes, etc. . .

are achieved better results than in the feature-inferred methods based on Neural Networks. They are mainly focused on the distributed representation as char/word-level embeddings. Finally, Li et al. [89] concentrate their efforts in reviewing the actual trend in this field: different kinds of deep learning techniques in NER for the contextualized representation of words, being one of the main approaches BERT [46].

In biomedical reviews, the tendency holds the same that in the general domain. In Goulart et al. [56] the retrieved documents in the review are tagged following a schema with different sorts of classifications such as the corpus or the features used. Once this tagging has been done, a numerical analysis is done with the retrieved documents concluding that the vast majority of NERs are being developed based on Machine Learning methods (92%). In Campos et al. [35], since most recent advances in NERs were coming from the application of Machine Learning methods, they just performed the survey on these methods. This make some references to prior methodologies such as hand-crafted rules but their main focus is on Machine Learning. They also give an overview of the features underlying the text which could be used as input for the models and study both commercial and open-source tools which implement those features and models. Like in the general domain, it is concluded that at this time CRF was the most widely and beneficial method used. In Wang et al. [136], six tools that implement different NER methods are studied and compared giving some final recommendations about what tools are more suitable for each entity types. These studies can be corroborate the Machine learning trend and the dominance of CRF within it. In Perera et al. [108], more recent advances such as Deep Learning approaches are introduced. From more traditional recurrent models such as BiLSTM [119] to more recent advances with the usage of Transformer-based models [132] such as BERT.

2.2.3 Data pipeline

NER task usually follows the pipeline showed in Fig. 2.1 in which the text is firstly pre-processed depending on the requirements of subsequent processes. Afterwards, a representation of the words which compose a text span is made which serves as an input to a NER model which performs the classification of these features to assign tags to the proper words. Sometimes, in order to refine the results, a post-processing step is also finally carried out. In the following sections, each of this steps has been revised in addition to an state-of-the-art review of current methodologies in BioNER modelling step.

2.2.3.1 Pre-processing

The first step we need to make when we are working with an NLP system is to work with meaningful units of text. This is usually achieved by firstly segmenting the text into the sentences in which it is compound and those sentences into words which are commonly known as tokens. In order to achieve that, some pre-processing steps have to be done:

- **Sentence Splitting:** It is the splitting of a piece of text into its sentences where each of them provides local contextual information for each of the words in which it is compound.
- **Tokenization:** It is the partition of a sentence into its subunits; i.e, commonly

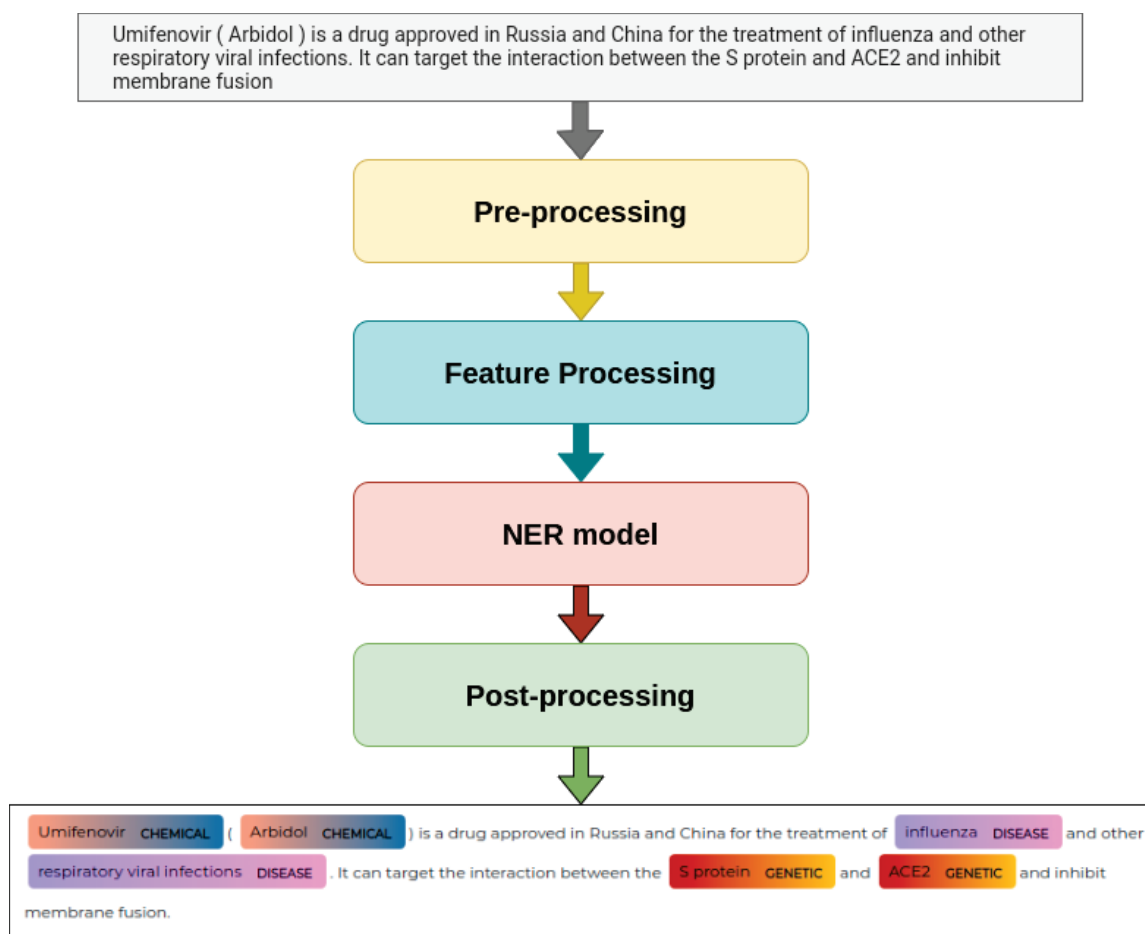


Figure 2.1: Pipeline widely used in NER with a text span example.

into words, which are known as tokens. It is an essential step since subsequent steps will be usually based on this and consequently, errors in this process will result in cascade errors in the following processing.

Nevertheless, these are not the only techniques in pre-processing. Depending on the subsequently used technique, we also could require to perform a Part Of Speech (POS) tagging in which we perform a grammatical tagging marking up the grammatical category of each word in a sentence. Sometimes it is also required some cleaning and normalization processes to try to reduce the polysemy in the text of some words. This includes tasks like stopword removal or case normalization for cleaning and lemmatization or stemming for normalization. Moreover, in specific domain NLP processes like the biomedical domain, it could also be necessary to perform some extra steps like resolving head noun sentences or abbreviation resolution.

Since most of these are widely used steps, lots of toolkits have been developed implementing these steps and further processing steps. One widely used kind of tools are the ones integrated in libraries like NLTK [91] in Python which is a toolkit that integrates different algorithms for NLP tasks like sentence splitting, tokenization, POS Tagging, etc. . . Another widely used tool is the one developed by Stanford University known as Stanford CoreNLP Toolkit [94] which covers the most common NLP tasks. With the increasing development of Deep Learning techniques, Neural-based tokeniz-

ers have also arisen, which is the case of the Google tokenizer SentencePiece [76] or WordPiece [118].

2.2.3.2 Feature Processing

A word representation is a mathematical object attached to a word. This has historically been one of the key points in NLP since subsequent tasks make use of the way we represent words. Features are attributes that describe characteristics of words or text spans and therefore the use of some of these features constitute a word representation approach. They are usually intended to serve as the input to algorithms, establishing before a feature vector which is referred as the set of each of the features regarding some token within a text and its values could be numeric, nominal or Boolean values. Depending on how we approach the problem of obtaining these features, we will distinguish between the approaches where these features are obtained in a more hand-crafted way through a Feature Engineering task and approaches where the job is done in a more automated way in the form of embeddings. In the following sections, both approaches and how they are usually performed are discussed.

Handcrafted Features

Earlier systems had the necessity of performing a feature engineering process to address the different features underlying each of the words in which a text is composed. It was needed to establish which set of features could behave better for the representation of the characteristics of each of these words. The objective of this was to establish the inputs for some kind of algorithm consumption, aiming in the NER case to perform the better classification process of the features of each word. Therefore, in this kind of approaches, it is vital to perform a good feature selection and detection since subsequent classification methods will be based on these data.

In Nadeau and Sekine [100] it is performed a classification of the different features usually used in NER which can be divided in:

- **Word-level features:** These are intrinsic to the word and therefore are related to how the words are composed. These features describe certain characteristics such as the presence of a numerical value or some kind of special character, capitalization, the presence of certain characters or prefixes/suffixes, the length, etc. . .

Focusing this category in a more linguistic way, then we could subdivide these features based on a linguistic function criterion:

- **Orthographic:** Characteristics about the composition of a word. Features related to the search for certain characters, symbols, and digits or the number of them.
- **Morphological:** Structures or sequences of characters observed within a word. Features related to the appearance of prefixes/suffixes, subsequence of n-characters (Char n-grams), and word shape patterns.
- **Semantic:** Characteristics about the token itself, its meaning. It is usually achieved by the tagging of the token in the text as an encoding with which we aim to associate the same words with the same encoding. In some

cases, we could have an interest in applying a normalization step in the pre-processing in order to group equivalent words such as the different tenses of the verb.

- **Syntactical:** Characteristics about the contextual relation between words in a sentence and the grammatical category to which they belong to. If we were interested in these syntactical features, then a POS tagging could be performed as a pre-processing step of our text.
- **List lookup features:** Features related to the presence of a word in a certain kind of list which is often referred to as dictionary, lexicon or gazetteer. The logic underlying these features is that if, for instance, we have the word "Remdesivir" and we check that this word is contained in a drug list, then we could infer that the probability of this word to be a drug is high. Nevertheless, we should take care of the polysemy which may occur in certain words that could appear in several different sorts of lists. The incorporation of these lists means adding knowledge to our system about the domain in which we are incorporating our lists. In the case of the biomedical domain, we could use some in-domain lists for matching specific domain terms and giving them a tag used as a feature. In some cases, it could be interesting for us to establish not only exact matching but also *fuzzy matching*, relaxing the criteria below we match an entity allowing some lexical variations. For this purpose, we could use some sorts of edit-distance such as Levenshtein [87] or Jaro-Winkler [141] distances.
- **Document and Corpus features:** Features that include meta-information about certain characteristics of a document or its statistics. This kind of features could help us to solve some co-reference, co-occurrences or related issues due to the multiple occurrences of a word throughout the text. Some meta-information could also be extracted to give indicators of the kind of information that could appear in a certain part of the text.

The selection of some of these features would allow us to build feature vectors for sentences and it is a key point since further classification in feature-based methods will depend on these vectors. For example, if we have the sentence: "New SARS-CoV-2 variant appeared in Brazil" and we set the following features:

- **Capitalization:** Boolean value where we establish True if it is a capitalized word and False if it is not.
- **Number of word digits:** Numerical value of the number of digits within a word.
- **Word Length:** Numerical value of the length of a word.
- **Internal Punctuation:** Boolean value which is True if there are some punctuation marks within the word and False if there is not.

Then, the resulting feature vector should be:

New	SARS-CoV-2	variant	appeared	in	Brazil
[[True,0,3,False],	[True,1,8,True],	[False,0,5,False],	[False,0,8,False],	[False,0,2,False],	[True,0,6,False]]

Embeddings

Conventionally, One-Hot Encoding has been the method used for semantic word representation due to its simplicity. Nevertheless, this simplicity is responsible of some critical flaws which lack its usage in modern systems since new embedding solutions have emerged. These flaws are related to its data-sparse and non-existent semantic relation between word representations, which produces very high dimensional vectors very dependant on the corpus size. The way this problem was faced was through the use of the distributional hypothesis of words [48] [65] which establish that similar words have a similar context. The distributed representation would allow us to share feature knowledge between the word representations, which would allow us to solve the sparsity problem establishing a similar representation for words that have a semantic and syntactical related meaning.

Word embeddings are techniques used for word representation as a real-valued vector that encodes some useful information through a distributed representation. Following the advances which Deep Learning has experienced recently, these word embedding approaches have been widely developed and used to model words and their context and consequently have a key role in downstream tasks such as NER and NEN. With this technique, it is no longer necessary to make a feature engineering process in which the features are selected and captured, this method automatically carries out the representation of words.

One of the most recent advances in the embedding of words is the different word representation depending on the context. This means that the same word could be represented in a different way depending on the context in which it appears, helping to solve some polysemy problems in which we can find the same words with different meaning in different text spans. Based on that characteristic, a classification of the embeddings has been performed, dividing them into the ones that take into account this advantage and the ones that do not.

- **Non-contextualized Embeddings:** These kinds of embeddings do not take into account how a word could have different meaning depending on the context where it could appear, which results in the same word representation although the meaning is different. Therefore, the polysemy problem is not directly addressed by these kinds of embeddings. These were the first kind of word embeddings developed [32] and are the following, ordered by released date:
 - **Word2Vec:** this technique learns the word representations from a corpus in an unsupervised way, capturing the semantics of words throughout a two-layer neural network. There are two similar algorithms implemented within its architecture: Continuous Bag of Words (CBoW) and Continuous Skip-gram models [95][96]. The difference between them lies in how they consider the inputs and how they output the multidimensional vector:
 - * **CBoW** do not consider the order of the input words since it works with a Bag of Words as its proper name stands for. It predicts the target word windowing a certain number of words, obtaining as output a vector which represents that word.
 - * **Continuous Skip-gram** makes use of the target word to predict a vector that represents a set of surrounding words guessing a set of potential neighbouring word probabilities.

Thus, it can be inferred that although we use them with the same purpose, the way they achieve this is opposite and the selection of one of them will depend on the application requirements. The hidden layer weight matrix will compound, after the training, the word vector lookup table which constitutes the word embeddings. Therefore, in pretrained Word2Vec models for different tasks, this is the matrix of the specific weights with which word representations are initialized.

- **GloVe**: Pennington et al. [107] proposed GloVe embedding as a solution to one of the main problems of Word2Vec: it does not take into account the global context of the words, just the local context in a certain size window. GloVe solves this using some global corpus-wide statistics of the words using a global co-occurrence matrix which then is used to determine the relationship between words. Therefore, although in training some contextual information is taken into account for training a representation of words, in its further use the contextual information of each given word it is not considered. In other words, the contextual information which involves this model is just offered as part of the training but not in the representation of each of the given words in its use offering the same embedding in polysemy situations.
- **fastText**: this method arose as a solution for the Out of Vocabulary (OOV) problem that the former embeddings present. For this purpose, [33] implemented an evolution of Word2Vec Embeddings (CBoW) and its key point is the use of sub-word level information. Most parts of languages contains some sub-word level information such as prefixes, suffixes... i.e, its morphological information. Consequently, all this information composes the internal semantic of words. Therefore, the use of this level as our atomic tokens helps, on the one hand, to overcome the OOV problem by unpacking the words in a sub-level of information and on the other hand providing more information about the word. Moreover, many biomedical terms follow different morphological rules in the word-formation (e.g., *keratin* and *myosin*, a great part of protein names incorporate the suffix -in) and thus, incorporating this character-level information, the embeddings should capture more entity information.

These have been the most widely non-contextualized embeddings used in literature since they appeared [95]. One of the major advances of these approaches is that since these embeddings are obtained through the semantic information of words, then the subsequent vectors obtained with these embeddings shall have semantic characteristics within the vector space they represent. Similar concepts are near in this vector space (see Fig.2.2) and some linear substructures can be observed, such as the same distance between synonyms or between the same relations between words... These linear substructures are usually got through vector differences as it can be observed in Fig.2.2.

Since the original models were trained in general domain corpus such as Wikipedia⁸, Common Crawl⁹, Gigaword 5¹⁰, etc... some efforts have been done for doing this

⁸<https://dumps.wikimedia.org/enwiki/>

⁹<https://commoncrawl.org/the-data/>

¹⁰<https://catalog.ldc.upenn.edu/LDC2011T07>

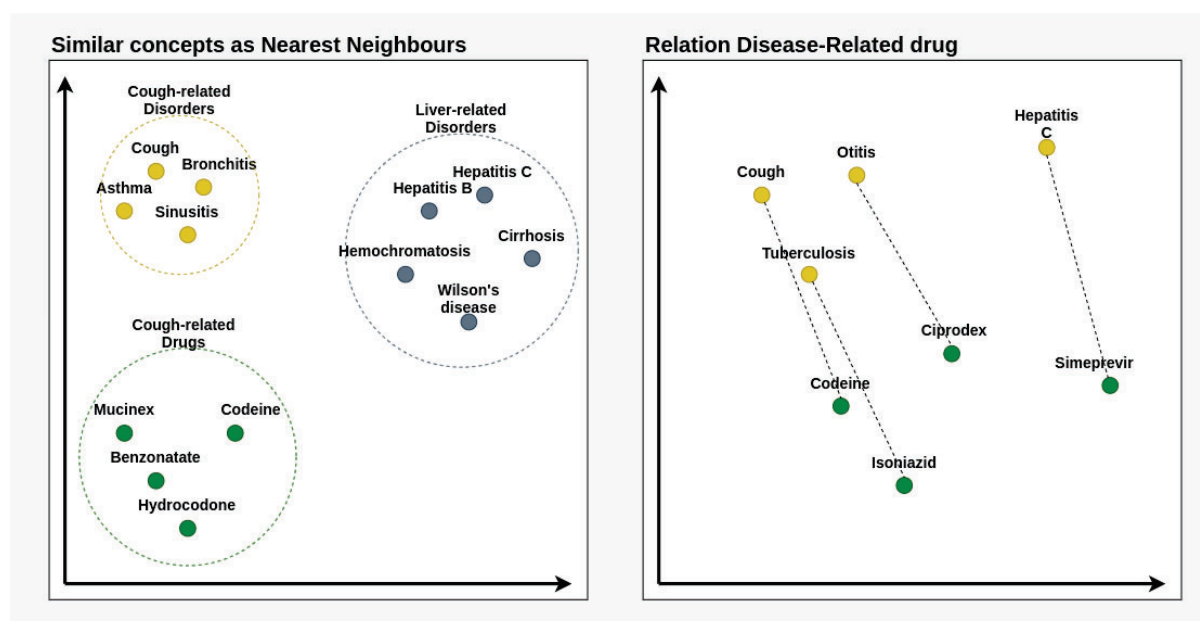


Figure 2.2: Visual example of some Vector Space properties for Word Embeddings in a bidimensional space for simplicity.

training in specific domain corpora such as the biomedical domain obtaining some biomedical word embeddings which behave better than general domain word embeddings in biomedical corpora tasks. These biomedical embeddings are usually augmented to represent clinical concepts in a more precise way. In Table 2.1 the biomedical embeddings implemented are shown.

Some publications extend the way this pretraining is achieved in the biomedical domain, incorporating different levels of information within these embeddings. Choi et al. [40] uses word2vec model to embed some codes from curated databases such as laboratory (LOINC¹²), drug (NDC¹³), disorder(ICD-11¹⁴) codes using a corpus which contain these codes within its records. This was proved to be useful in applications of these embeddings in clinical notes which generally make some reference to this kind of codes for standardization. Beam et al. [30] and De Vine et al. [45] apply a similar idea using medical controlled vocabularies from biomedical unified databases such as UMLS¹⁵ in order to provide more specific domain knowledge from the curated thesaurus. For this pretraining, they make use of the Concept Unique Identifiers (CUI) which groups into code some medical concepts with their synonyms and ambiguities. Therefore, in some cases the pretraining of these embeddings are not just extended to a biomedical corpus to achieve an unsupervised learning of embedding weights, but also it is extended in a more controlled way to use specific kinds of knowledge such as certain kinds of clinical codes, certain groups of semantic types or grouping together some concept ambiguities to try to catch this variability

¹¹<https://datadryad.org/stash/dataset/doi:10.5061/dryad.jp917>

¹²<https://loinc.org/>

¹³<https://www.fda.gov/drugs/drug-approvals-and-databases/national-drug-code-directory>

¹⁴<https://icd.who.int/browse11/l-m/en>

¹⁵<https://www.nlm.nih.gov/research/umls/index.html>

Publication	Model	Corpora/Concepts	$\ \mathbf{D}\ $
Chiu et al. [38]	word2vec	Pubmed, PMC	-
Choi et al. [40]	word2vec	ICD-9, NDC, LOINC, medical claims	300
		UMLS, Clinical Notes ¹¹	300
De Vine et al. [45]	word2vec	UMLS, medical journal abstracts	200
Kosmopoulos et al. [73]	word2vec	MEDLINE, Pubmed	200
			400
Moen and Ananiadou [98]	word2vec	PMC	200
		Pubmed	
		Pubmed, PMC	
		Wikipedia, Pubmed, PMC	
Beam et al. [30]	word2vec, GloVe	UMLS, Pubmed, medical claims	-
Gehrmann et al. [51]	word2vec	MIMIC-III	50
Flamholz et al. [49]	word2vec	PMC	100,300,600
	GloVe	MIMIC-III, Wikipedia	
	fastText	PMC	
Zhang et al. [147]	fastText	MesH, PubMed	200
Chen et al. [37]	fastText	MesH, PubMed	200
	sent2vec	PubMed, MIMIC-III	700

Table 2.1: Publicly available biomedical non-contextualized embeddings. $\|\mathbf{D}\|$ is referred to the dimension of the vector.

within embeddings. This means to specify the pretraining of the embeddings as much as our target task requires.

- **Contextualized Embeddings:** The challenge that former methods were facing was the fact that each term is represented by just a single vector, regardless of the fact that its meaning may change depending on the context in which it is used, i.e., polysemy could appear. Contextualized Embeddings appeared in order to solve this issue by generating different embeddings for the same word depending on the context in which it appears. Consequently, these techniques used a language model to solve the context issue. A language model consists in generating the maximum likelihood token given a sequence of n words and this is the task that is generally used in the unsupervised training of this kind of embeddings. The following are the most widely adopted Contextualized Embeddings:
 - **ELMo:** Peters et al. [109] proposed the first solution to polysemy providing embeddings dynamically taking into account the context of a word through the development of Embeddings from Language Models (ELMo). For this purpose, ELMo uses the internal states of multiple layers of Bi-LSTMs to take into account in the Language Model both forward and backward in a text. It uses character-level embeddings to address the OOV problem and to give word-formation information in the representation of each word. These character embeddings are passed through a convolutional and pooling layer to get a fixed-length representation of the word. The use of convolutional layers allows the model to capture more powerful representations of each word. Afterwards, this representation is passed through multiple

Bi-LSTM layers (originally 2 layers were proposed [109]) which give us the output word embedding in a way that it captures the forward and backward contextual information. Moreover, ELMo allows to fine-tune its word representation depending on the downstream task for which it will be used. For example, if we wanted to use ELMo for a BioNER task, then its model weights would be scaled in the training to try to optimize the performance of the model in BioNER.

- **GPT**: Radford et al. [114] proposed the Generative Pre-Training (GPT) model to address the polysemy problem through the incorporation of forward-ing Transformer decoders [132] for the feature extraction through a self-attention mechanism. In further years, this model was extended through the use of each time more massive corpora in training and other enhancements. In GPT-2 [115] 1,5 billion parameters were used (10x the data of GPT), GPT-3 [34] exponentially rose this number increasing the number of parameters to 175B. Both GPT-2 and GPT-3 source code have not been fully released since they became commercial-use models.
- **BERT**: Devlin et al. [46] proposed that former Contextualized Embedding models were not making good use of the contextual information. On the one hand, GPT just used forward contextual information, on the other hand, ELMo used forward and backward layers in the model but its use was through the superposition of both unidirectional models. Consequently, Devlin et al. [46] proposed Bidirectional Encoder Representations from Transformers (BERT), an improved model which takes into account the information from both directions at the same time. The implementation of this model used a bi-Transformer technique which through a self-attention mechanism aims to extract feature information.

Moreover, BERT incorporates further advances. In relation to the input used in the model, WordPiece[118] embeddings were used as the representations of each word which supposed an upgrade in relation to the previous character-level representations. WordPiece aims to represent words in a subword-level including, apart from characters, the most frequent combinations n-grams of characters/symbols/numbers in the vocabulary in which it was trained. This technique enhanced the richness of the information in each word formation and at the same time solved the OOV problem. Apart from this, the input in BERT includes a position embedding that encodes the position of each word and a sentence embedding that encodes the belonging of a word to a sentence for training the Next Sentence Prediction (NSP) task in which it is predicted whether a sentence follows another given. In the NSP task, the models aim to capture more long-term information in the text. Another advance they introduced was the use of a Masked Language Model (MLM), instead of a simple Language Model, which randomly masks a certain percentage of words in the text to train the model in an unsupervised way predicting the masked words. This differentiation with respect to the former approaches allows BERT to avoid the locality bias in which models are just focused on words that are closer to the target word in each moment. The training of BERT, both in MLM and NSP tasks, grants to understand how the relations between words and sentences are; i.e., local and long-term contextual information. Once the model has been trained on

these tasks, it can be fine-tuned for downstream tasks such as NER.

In Fig.2.3 a visual comparison of these models is shown. It is easily seen how these different models make use of text both as input and at different levels of its internal architecture.

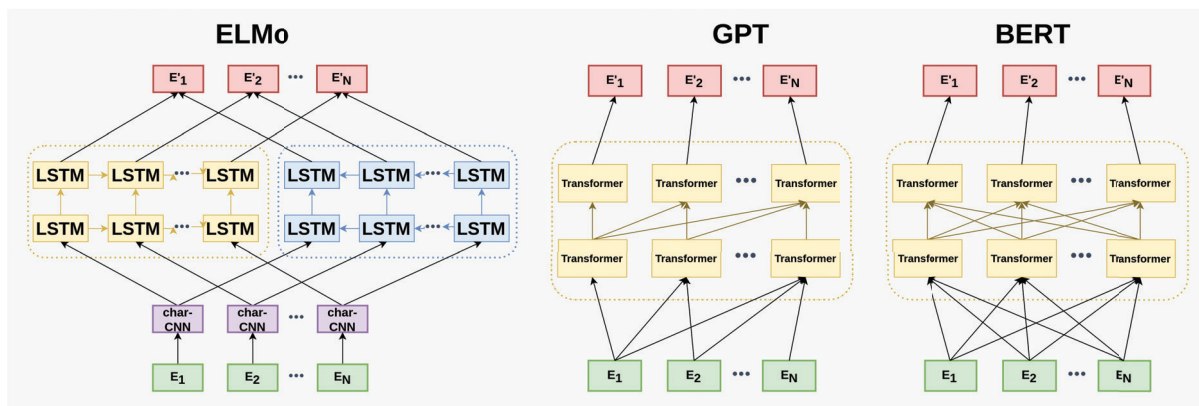


Figure 2.3: Simplified comparative of Contextualized Embedding models

The training of an embedding model is crucial to its success; initially, models like BERT were pretrained in general corpora such as BookCorpus and Wikipedia. This helps the model to do well on general text, but in extremely specialized contexts, such as the biomedical domain, the accuracy of the model could suffer as a result of the inability of the model to adequately manage domain information. As a result, a solution to this issue is to do pretraining on an in-domain corpus [63], allowing the model to integrate domain information as part of the model. This has been performed in biomedical corpora, resulting in Contextualized Embedding models that are well-suited to the biomedical domain and work better than general domain models managing this kind of information. Since the Contextualized Embedding model which better results has been demonstrated to achieve is BERT, the efforts in pretraining, which is a very high intensive computationally task, have been focused on this model (See Fig. 2.4).

Different pretrained models have been obtained in biomedical and related domains as the following ones:

Lee et al. [84] proposed BioBERT, a BERT model pretrained on large biomedical corpora, which has been the biomedical pretrained model with more impact since its performance improved previous state-of-the-art results in most of the biomedical tasks in NLP field. This model, whose weights were initialized following the original BERT model trained in the general domain (Wikipedia and BookCorpus), was pretrained in large-scale biomedical corpora with PubMed Abstracts and PMC full-text articles.

Alsentzer et al. [24] followed the idea of pretraining a BERT model, proposing ClinicalBERT that uses in pretraining MIMIC-III clinical corpus which contains approximately 2 million clinical notes. Two kinds of models were tested: one pretrained from the original BERT and another from BioBERT showing a better behaviour in clinical data the one initialized from BioBERT. However, it should

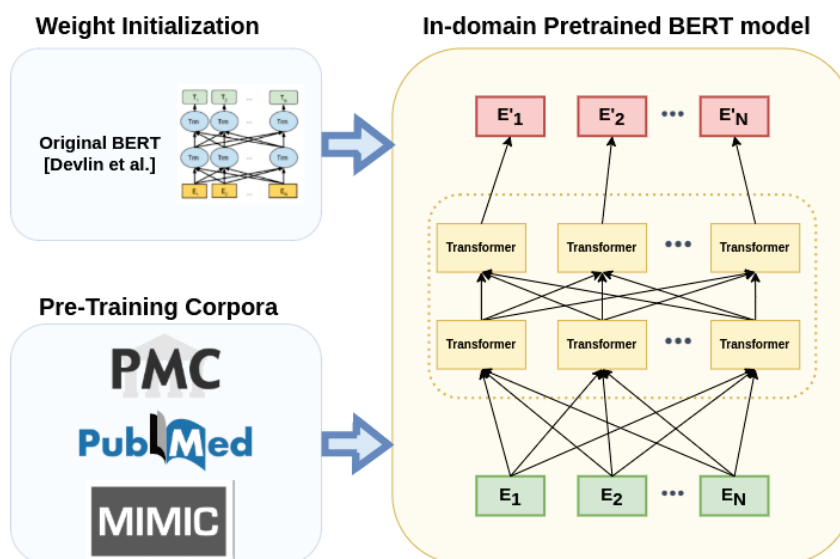


Figure 2.4: Pre-training process from multiple sources

be noted that clinical text is not exactly the same than the text found in biomedical articles. Style, form and focus of linguistic expressions in both kinds of texts are substantially different, making that despite the fact that both belong to the health domain, models specifically pretrained on clinical corpora will better perform on clinical text and models pretrained on biomedical articles corpus will better perform on this kind of publications. Therefore, even though it is an almost identical domain, the more domain-specific is the pretraining in the target data in which we aim to perform the task, the better results this model will have.

Beltagy et al. [31] also performed a pre-training in a BERT model called SciBERT using 1.14M Semantic-Scholar¹⁶ scientific papers in which most of them were from biomedical domain and a little part were from computer science domain. Depending on the aim of the application of this model, the combination of different sorts of domains for pretraining could be beneficial or not since it is a more general model than the previous ones which just focused on one narrow domain.

Peng et al. [106] developed multiple pretrained models called BlueBERT based on the original BERT initialization weights. This pretraining makes use of different combinations of PubMed abstracts and MIMIC-III notes, mixing biomedical and clinical domain data. In the experimental results, it was identified that in NER task on biomedical texts, the inclusion of clinical data on pretraining(MIMIC-II) worsened the results. Therefore, it is concluded the same than in ClinicalBERT [24], the more specific is the pretraining in relation to the target task, the better results are obtained. Nevertheless, care must be taken since if different tasks or a combination of NER with other tasks is proposed in a NLP system, perhaps the election of a model pretrained on a more general domain as it is the case of biomedical plus clinical (PubMed + MIMIC-III) could benefit the overall behaviour of this NLP system.

¹⁶<https://www.semanticscholar.org/>

Gu et al. [61] proposes to carry out a pretraining from scratch, stating that initializing the model from general domain in pretraining could hinder the pre-trained model performance. It has been previously assumed that domain-specific pretraining may benefit from continuous pretraining from general domains. Nevertheless, in some domains like biomedical, the general domain text is significantly different, raising the prospect of the negative effect of the use of general domain knowledge in pretrained model performance. Therefore, PubMedBERT model is proposed using purely biomedical text (PubMed), initializing BERT model from scratch. This initialization makes it necessary to build a vocabulary for the model. Since in this model only biomedical texts are used, this vocabulary contains a larger set of biomedical terms than the ones that used Original BERT vocabulary. Therefore, in target tasks in the biomedical domain, this model benefits from having been trained just in biomedical texts, achieving a more precise representation of the biomedical knowledge underlying a piece of biomedical text.

In Table 2.2 a comparative of the Contextualized Embeddings is done. Corpus used in pretraining is compared, establishing between square brackets the corpus taken for initialization, which in the case of [Wikipedia + BookCorpus] is referred to the initialization from the original BERT from Devlin et al. [46] and in the case of [Wikipedia + BookCorpus + PMC + PubMed] is referred to BioBERT [84]. Biggest pre-training is achieved by BioBERT which uses larger corpus in pre-training (PubMed + PMC). Just two proposed models chose pretraining from scratch (PubMedBERT and one of SciBERT models) having to build the vocabulary for Wordpiece tokenizer from scratch, obtaining a vocabulary based on the texts in which it was trained. The other models used preexistent vocabulary from general domain corpora obtained by training in Wikipedia and BookCorpus which Original BERT achieved.

Year	Model Name	Vocabulary	Pre-Training corpus	Text Size (words)
2018	BERT [46]	Wikipedia + Books	Wikipedia + BookCorpus	3,3B
2019	BioBERT [84]	Wikipedia + Books	[Wikipedia + BookCorpus] + PMC	4,5B
			[Wikipedia + BookCorpus] + PubMed	13,5B
			[Wikipedia + BookCorpus] + PMC + PubMed	18B
2019	ClinicalBERT [24]	Wikipedia + Books	[Wikipedia + BookCorpus + PMC + PubMed] + MIMIC-III	0,5B
			[Wikipedia + BookCorpus] + MIMIC-III	
2019	SciBERT [31]	PMC + C.S.	Biomedical & C.S. Semantic-Scholar papers	3,2B
		Wikipedia + Books	[Wikipedia + BookCorpus] + Biomedical & C.S Semantic-Scholar papers	
2019	BlueBERT [106]	Wikipedia + Books	[Wikipedia + BookCorpus] + PubMed	4B
			[Wikipedia + BookCorpus] + PubMed + MIMIC-III	4,5B
2020	PubMedBERT [61]	PubMed	PubMed	3,1B

Table 2.2: Biomedical pre-trained BERT models

2.2.3.3 BioNER Models

BioNER is intended as a classification task in which we use the features previously obtained to make a prediction of the entity tag which belongs to each of the

words within a given text. Therefore, BioNER models make use of the different pre-processing and featurization techniques formerly exposed and depending on how is the subsequent use of these features and how is the architecture of the model, we have different sorts of BioNER methods. Methods employed in BioNER can be classified in three categories: Rule/Dictionary-based, Machine Learning and Hybrid Models approaches¹⁷. These categories are the most widely used in the literature, but nonetheless rule- and dictionary-based models usually appear split in two different approaches. Because of they are similar and just earlier approaches applied them separately, they have been grouped together. In Machine Learning approaches, a subdivision can be established between supervised learning and unsupervised learning. In supervised learning we can find another subclassification depending on the paradigm, in which we can find traditional ML approaches and Deep Learning, the currently most widely used approach. Due to the high extension and the different paradigm which Deep Learning models suppose, these are considered independently to the traditional ML models which are taken into account in the supervised learning section. Lastly, Hybrid Models use a combination of different approaches to leverage benefits from different methods.

Rule- and Dictionary-based approaches

These were the earlier approaches that appeared in NER field. They make use of the design and implementation of handcrafted rules in order to categorize the named entities in text based on the features that have been previously captured. Therefore, a set of rules are designed to describe how are the patterns of the different sorts of named entities to characterize them. Consequently, a great knowledge about the working domain is required in addition to linguistic expertise to correctly design that set of rules in a proper way, which is also a very costly and time-consuming labour. It is important to note that the implementation of these approaches sometimes takes directly the features as text format without the need for representing it as a feature real valued vector.

The features these methods are usually focused on capturing were the morphological and orthographic features. For instance, a word with suffix *-in* is usually a protein name (e.g., keratin, albumin, etc. . .) but nevertheless, another rule within this rule will have to be set since there are several drug names that usually use this suffix (e.g., lidocain). Consequently, the main drawback of this approach is the requirement of an effective capturing of all possible patterns which can occur in a given entity and the time-consuming process this handcrafting design involves.

It can be then inferred that rule-based approaches require high cost work since lots of rules have to be set to aim to have high precision in the classification. Hence, these approaches are usually just focused on very defined narrow areas in order to try to cover well the intrinsic rules within the entities in this area. This kind of approaches are usually combined with **dictionary-based approaches** which make use of large databases of named entities to trigger some matches between our text and the database. This is a good way of adding some specific domain knowledge to our model like the biomedical knowledge with the inclusion of biomedical dictionaries.

¹⁷The approaches which make use of dictionary in the phase of featurization were not considered as hybrid models. Hybrid models were considered if the dictionary was used just as part of the modelling phase.

Since some morphological variations may not be collected in some dictionaries, some fuzzy matches are usually used. The use of dictionaries usually raises the recall whether the selected dictionaries and the terms within them fit good to our purpose since these dictionaries can cover the appearance of lots of entities which can appear in text. Nevertheless, not covered terms and new terms will not be captured with just this method, and that is the main reason why rule and dictionary-based approaches are often applied jointly, as they are supplementary.

As I have already mentioned, earlier approaches were based on these methods. In recent years, the number of BioNER methods that use these approaches has dramatically fallen because of its important drawbacks and the appearance of different state-of-the-art methods, which also ease the model implementation. In this review, just the following publication was retrieved since in recent years the efforts of developing BioNER solutions have been focused on more advanced approaches or on hybrid approaches that use dictionary-based methods in conjunction with machine learning methods [116].

Pafilis et al. [105] developed a dictionary-based approach for the recognition and normalization of organisms and species in text. Its main focus was set on developing an updated and large dictionary initialized from the NCBI Taxonomy¹⁸ where some variants were further added. The way this dictionary was implemented in the search of entities in text was through the use of hash tables, which in addition allowed some orthographic variations. Some expansion and post-processing rules were also set to look for a result improvement.

Machine Learning approaches

Previous approaches relied on the composition of handcrafted complex rules which were costly, time-consuming and require expertise knowledge in the target domain for developing the rules which better fit the target entity classes. Furthermore, these implemented rules were not reusable since they are highly adapted to an entity class.

Machine Learning approaches offer a solution to this rule design generation problem through the automatic pattern identification which Machine Learning methods statistically obtain. This solves the main problems which rule systems present through the automatic establishment of patterns in a given set of data. The growing number of available data on the Internet has boosted the number of resources offered for the development of those kinds of methods, downplaying the importance of the data availability drawback in fields like the biomedical domain. Commonly, Machine Learning approaches have been widely divided attending to the labelling which the given training set of data presents: Unsupervised Learning does not make use of labelling unlike Supervised Approaches.

Unsupervised Learning approaches

Unlike supervised approaches, and as with rule- and dictionary-based methods, unsupervised learning approaches do not require the use of labelled training data to develop the model. This is one of the main advantages of both sorts of approaches since not always we can provide data labels for the training of our model. Nevertheless, increasing resources are appearing in the biomedical domain, making that

¹⁸<https://www.ncbi.nlm.nih.gov/taxonomy>

the necessity of labelling not a big problem. Consequently, the role of unsupervised approaches in more recent literature has fallen, being not very often this choice for modelling BioNER. Most unsupervised methods use applying a clustering for word classification based on some assumptions such as that the sense of a word is usually influenced by the words that surround it. Despite the great advantage of not requiring annotated datasets, the results are far below from the expected from other kinds of models.

Zhang and Elhadad [146] developed an unsupervised solution that used three steps to address the NER task. In the first step, they make use of the *seed term* concept with which a dictionary for each of the target entities is made from the semantic groups, types and concepts from UMLS. In the second step, boundaries of entities candidates are detected through chunking with an IDF filtering of Noun Phrases. Finally, a classification is done using the idea of entities of the same class usually have similar vocabulary on context. For this purpose, a *signature vector* is generated using the TF*IDF concept, where this signature is calculated for each entity class as the average of the *vector signatures* of all the seed terms of that class that appear in the corpus. Once, this has been done a cosine similarity metric is calculated for each of the candidates obtained in step 2 with each of the *signature vectors*. The entity is classified as the class with a higher score if a determined threshold is passed (if not, it is not considered an entity). The results obtained in the GENIA corpus were substantially lower than state-of-the-art approaches, obtaining acceptable results in protein recognition with an F1-score of 67,2 and very poor results in cell line recognition with 19.9.

Supervised Learning approaches

These approaches use the handcrafted features exposed in the section 2.2.3.2 which are used as input for a machine learning model which through the use of annotated training instances discovers hidden patterns within the data which shall constitute the underlying classification rules to tag a token as an entity or not. Therefore, the previously observed featured training data is modelled to create a probabilistic description of the entity classes in which we want to perform the classification NER task to make an appropriate response to unseen data. The use of training data is essential in supervised learning approaches and consequently, the performance of the ML model will broadly depend on the quality and magnitude of the employed dataset in the training phase.

There are several Supervised Machine Learning methods that can satisfy the NER task. In the earlier stage of Machine Learning applied to the NER task, the most frequent models were Hidden Markov Models (HMM), Maximum Entropy Markov Models (MEMMs) and Support Vector Machines. Nevertheless, the appearance of the Conditional Random Fields (CRFs) [78] soon became CRF the most widely used model in supervised approaches. This was because CRF presented some advantages over the problems which previous methods had: it overcame the label bias problem [78] which some Markov models such as MEMMs presented, it also downplayed the importance of the independence assumptions required in the parameter learning of HMM and it was proved to be less computer-intensive in training than SVMs. These advantages involved that the trend in later years was dominated by the use of CRF in different implementations for the NER task:

Campos et al. [36] employs an intensive feature engineering process to obtain different sorts of features which better describe tokens in different aspects such as morphological, dictionary matching, local context, etc. . . The obtained features are used as the input to multiple first- and second-order CRF models parsed both forward and backward. The results from these models follow a combination strategy based on confidence scores in which the tagging of an entity depends on the confidence value given by CRF models.

Wei et al. [138] designed a high specific solution for the recognition of genetic variants developing a set of features taking in mind the peculiarities of these entities, designing for that purpose dictionary, linguistic, character, semantic, case pattern and contextual features. Some of which make use of regular expressions to capture the underlying characteristics of genetic variant entities. The resultant feature vector is used in a CRF model which after the training phase predicts the tag for a given token. Wei et al. [139] extended tmVar with a second version, tmVar 2.0, which integrates a normalization process to the variant entities found.

Leaman et al. [83] make use of different CRF models which differ in features used, parameters and implementations of the model, and post-processing variations. The feature set used is highly specific for chemical entities and uses a broad number of characteristics underlying the chemical names. Results from both models are combined following three different kinds of rules to obtain the corresponding tag for the given tokens.

Leaman and Lu [81] stated that an often use of the tagged entities given by NER was the normalization of these entities. The most usual approach was to apply this normalization step following the entity recognition which could cause cascading errors if the tagging was mistaken. Leaman and Lu [81] addresses this problem facing both tasks at the same time using a semi-Markov structured linear classifier which combines the use of handcrafted features for NER and supervised semantic indexing based on Tf*idf normalized vector for normalization. The combination of both tasks was done obtaining a score by applying the model both recognition and normalization and adding both scores.

In Table 2.3, a comparative review is done for the supervised approaches previously overviewed papers which were taken as relevant in our review. The set of input features considered for each of the approaches is shown. It should be noted that some of the features mentioned in the articles have been grouped for simplification and comparison. As it was seen in Section 2.2.3.2, Nadeau and Sekine [100] performed a widely used classification of these features, based on which features were grouped. In *Others*, features that can not be grouped are found.

As it can be inferred, the design of a suitable set of features is a key point in this kind of methods, since the success of further models broadly depends on how are these features. Moreover, a good model implementation also depends on the training corpus used for the correct parameter learning of this model. Therefore, both requirements have to be satisfied to expect optimal results. Because of this, the features will have to be designed following the underlying characteristics of an entity and the dataset used will have to be adapted just to the entity we designed the features. Consequently, the design of supervised models results in high specific models which are hardly

¹⁹Tool prepared to be adapted to several entity types. Nevertheless, in its publication it was probed with disease and chemical entities given by its training corpus

Reference		Campos et al. [36]	Wei et al. [138]	Leaman et al. [83]		Leaman and Lu [81]
Year		2013	2013	2015		2016
Model type		CRF	CRF	CRF first order	CRF second order	Semi-Markov Linear Classifier
Input Features	Orthographic	✓	✓	✓	✓	✓
	Morphological	✓	✓	✓	✓	✓
	Semantic		✓		✓	✓
	Syntactical	✓	✓	✓		✓
	List lookup	✓	✓	✓		
	Local Context	✓	✓		✓	
	Others	-	-	ChemSpot [116] result		TF*IDF normalized vector
Training corpus		GENETAG, JNLPBA	tmVar Corpus, MutationFinder corpus	BC4CHEMD		B5CDR, NCBI disease
Entities		DNA, RNA, gene, protein, cell line, cell type	Genetic variations	Chemicals		Non entity specific ¹⁹

Table 2.3: Revised Supervised BioNER models

adaptable to other entities. This drawback, which rule-based systems have already presented, is one of the main problems which subsequent approaches face.

Deep Learning approaches

Deep Learning term, which is a subset of Machine Learning, is used to make reference to the use of Neural Networks with many hidden layers and nodes forming a deep network. This deepness allows those models to catch complex nonlinear patterns within the training data and use these patterns for later inferring tasks. The evolution and development of computational power have boosted the use of these kinds of models, resulting in increasingly complex and deeper models. NER task has also been affected by the growing popularity of Deep Learning Models, currently replacing former approaches such as CRFs, which only used linear patterns in data. Furthermore, the former approaches were based on the use of different kinds of specific features underlying the target text. This generally made that the approaches were entity-specific since feature vectors were designed based on that entity, making its development costly and hardly reusable with another entities. Moreover, some of these approaches were also highly optimized for specific GSCs, making that features within these approaches are also designed specifically for these corpora, further burdening the cost and reusability. Although the first Neural Network proposal for NER [41] employed these handcrafted features, soon subsequent methods used word embeddings solving some of the main problems presented by the handcrafted features such as reusability and design cost. Word embeddings made possible to represent text tokens in a distributional way with linguistic features underlying this representation. As seen in Section 2.2.3.2, there are several models which achieve implementations of word embeddings, allowing the authors to design models using some of these different word embeddings. Depending on the kind of embedding used for the implementation of the NER model, we can find two kinds of models attend-

ing to the fact whether these embeddings are non-contextualized or contextualized, a classification previously made in Section 2.2.3.2.

Non-Contextualized Embeddings Models

Models which are designed using non-contextualized embeddings generally use two main kinds of Neural Networks. On the one side, Recurrent Neural Networks (RNNs), which are a kind of Neural Network mainly adapted to sequential data such as text, thanks to the employment of a temporal behaviour using internal states within the network nodes forming a directed graph as a way of granting it some memory of the former states. The most widely known RNNs are LSTMs [67] and GRU [39] which were developed in order to solve the vanishing gradient problem²⁰ which showed RNNs presented. Moreover, the solution to this problem grants the model to have long-term memory enhancing the contextual information it can manage. This is achieved through the use of different sorts of the called "gates" which are neural networks used for controlling the flow of information within the network through the learning of the important inputs in the sequence storing their information in a memory cell. GRU follows the same idea than LSTMs but using less kinds of gates obtaining similar results in practice with a lower computational cost. It is also a common practice, to use Bidirectional LSTMs (BiLSTM) [58] to capture inputs both in left-to-right and right-to-left direction to encode dependencies on elements from both directions. In Fig. 2.5 a visual comparison between these main models is shown. Lample et al. [79] first proposed the use of these architectures in NER obtaining state-of-the-art results in a general corpus with a Bidirectional LSTM network followed by a CRF layer (BiLSTM - CRF). For this purpose, character-based word representations learned from the supervised corpus and unsupervised word representations learned from unannotated corpora were used.

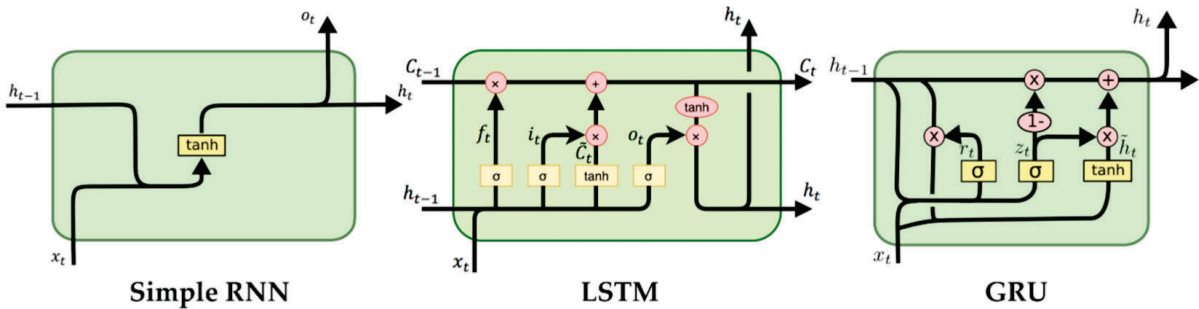


Figure 2.5: Visual Comparative of most widely used RNNs units. Image obtained from Aslam et al. [26]

On the other side, Convolutional Neural Networks (CNNs) have also been proposed as a means of tackling NLP tasks. CNNs are a kind of Neural Network designed mirroring human visual cortex function through the inclusion of several specialized layers following a hierarchy. The main kind of layers used are

²⁰Problem found in gradient-based learning methods which exponentially decreases the gradient used in weight updating in the training stopping the model from further training.

the convolutional layers which make use of kernel filters²¹ in convolutional operations, these are structured following a hierarchy where firsts layers obtain simple and generic features and deeper layers find more complex and specific features. One of the main characteristics of CNNs achieved by its design is its spatial expertise which makes them very beneficial to abroad Computer Vision problems. Nonetheless, this networks can be adapted for using them in text. While images are 2-dimensional or 3-dimensional, text is 1-dimensional and consequently convolutions are applied just in one dimension. The time dimension attached to text sequences may be treated as if it were a spatial dimension allowing the network to recognise local patterns in text sequences as it does with the recognition of patterns in images. Moreover, these patterns learned in a certain location may be identified in a different location grating the CNN to learn about the word morphology. Collobert and Weston [41] first proposed the use of a CNN architecture to accomplish a multitask problem where NER is one of these tasks. CNNs have been broadly used as a way of creating char embeddings because of its power to learn and identify word morphology patterns. In Fig. 2.6 a common structure of CNN models applied to NLP is shown.

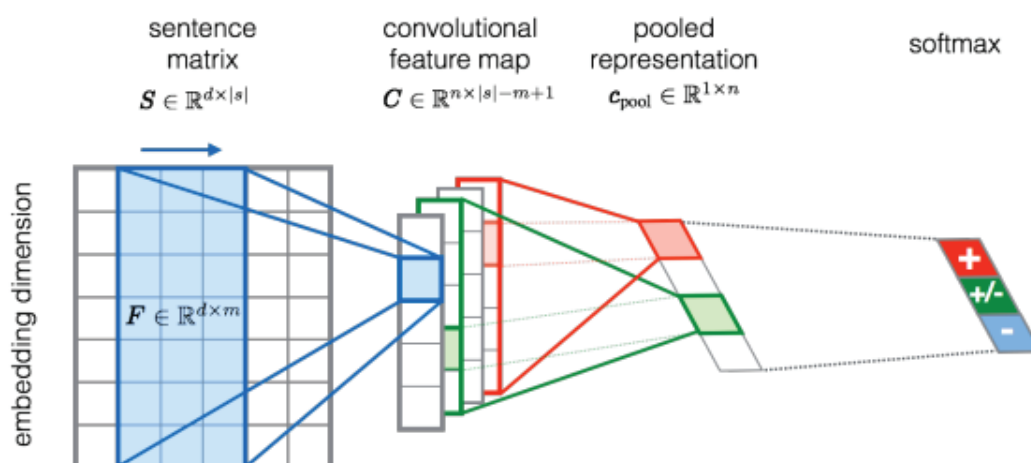


Figure 2.6: General structure followed by CNN models applied to NLP

The methods retrieved in the review that follow some of these models are the following ones:

Habibi et al. [64] implemented the model firstly proposed by Lample et al. [79], adapting a Word Embedding + BiLSTM-CRF model to the biomedical field in a BioNER task, obtaining state-of-the-art results in most GSC tests overcoming previous widely used methods such as CRFs. In Figure 2.7 the structure of this model can be observed. The first layer is an embedding layer in which its embeddings are the concatenation of two kinds of embeddings: a word-level embedding obtained from pretrained word embeddings [98] in a lookup-table and a character-level embedding obtained by applying a Bi-LSTM to the sequence of characters of each word. The second layer is a Bi-LSTM layer which aims to get a refined and partially contextualized representation of its input to serve as the final input, a CRF-layer which obtains the final output: the token tags. The

²¹These are the applied matrix of values in the mathematical operation of convolution and the weights within them are the values to optimize in training.

combination of these layers makes it possible to train the entire model together by backpropagation.

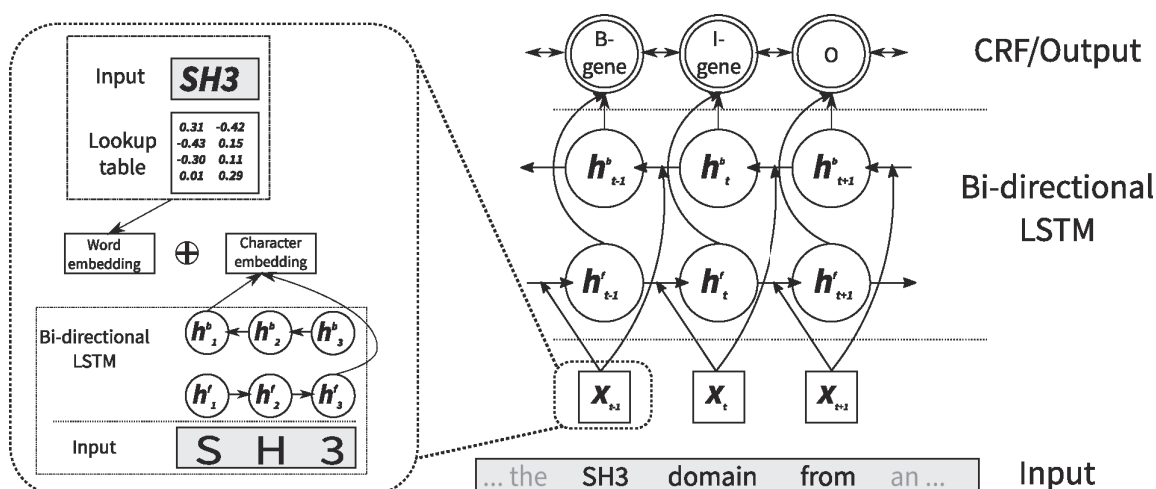


Figure 2.7: Most widely used BiLSTM-CRF model since its first proposal by Lample et al. [79]. Image obtained from Habibi et al. [64]

Crichton et al. [43] implements a multitask CNN model which aims to leverage what they have learned in an auxiliary task to improve its performance in a similar task. There are several BioNER training sets which are focused on the same entity class and therefore, several of them may be used to train a model for a certain entity instead of just using a single dataset for a certain entity. For this purpose, two CNN models are implemented: the first one shares the CNN layer and just the top fully-connected layer is changed between tasks and the second one which uses two parallel models in which one model is used in an auxiliary task such as POS tagging to give to the main task model useful information at the fully-connected layer level.

Gridach [59] studies the importance of a Character-level embedding input in a BiLSTM-CRF model. It states that Character-level embeddings have been probed to improve performance in several kinds of tasks including NER and could be useful in fields with very complex words such as biomedical thanks to the OOV problem solving. OOV problem was seen to be a problem found in a large part of Word Embedding pretrained models such as Word2Vec, and therefore the inclusion of Character Embeddings could help to overcome the OOV problem in the use of these. To add Character-level representations, a BiLSTM model is used to compute the character-based vector embeddings of words. Afterwards, these character embedding vectors are concatenated with a pretrained word embedding vector. The resultant model is the same it was shown in Fig. 2.7. It is also tested the importance of using different pretrained word embeddings, concluding that biomedical word embeddings [98] outperforms better than general domain embeddings.

Lyu et al. [93] employs a BiLSTM-CRF model with character and word-level embeddings as input. Character-level inputs are obtained from an attention-based model and Word-level embeddings are obtained from pretrained embeddings and from its own training. The comparison between the different word embeddings results in a better performance in biomedical fine-tuned word2vec embeddings

(its own training). Moreover, the inclusion of Character-level embeddings also improves performance of the model.

Unanue et al. [130] explores how the use of different kinds of outputs can affect the performance of a BiLSTM-CRF model. It tries with a concatenation of different kinds of word embeddings, a character-level embedding, and the inclusion of handcrafted features as input. It concludes that retraining general domain GloVe embedding in a biomedical corpus, in this case MIMIC-III, enhances the performance of the model, character-level embeddings also improves the model performance but handcrafted features inclusion, not only increases the cost of development but it lowers results.

Luo et al. [92] states that in general previous efforts in NER were focused on a sentence-level processing of texts. Therefore, dependency between taggings in different sentences is not considered leading to tagging inconsistency problems in which same mentions in a document are tagged with different labels. In order to solve this problem, an attention layer is used forming a BiLSTM-Att-CRF model in which the attention technique aims to capture similar entity attention at the document level. Moreover, Luo et al. [92] proposes to use some additional handcrafted features as input, using some syntactical information such as POS tagging and chunking and some dictionary-based features. Its experimental results show that the attendance layer improves the results, proving to be effective to mitigate the tagging inconsistency problem. The additional handcrafted-features barely improve the results and therefore it is concluded that it is not worth the substantial time-consuming effort it involves.

Ju et al. [69] exposes that multiple named entities contain *nested entities*, which are embedded entities which are included in another entities. Most previous systems does not take into account this problem focusing just on non-nested entities, also called *flat entities*. Therefore, a solution to this problem is proposed using a stacked layer model in which the number of layers is adjusted dynamically to the level of entity nesting. This model, based on LSTM-CRF stacked layers, tackles the problem extracting entities from inside to outside, sharing parameters among the different BiLSTM and CRF layers, in such a way that a layer output is used as input to the the next layer until non entities are found.

Zhu et al. [150] states that biomedical texts are usually composed of very long sentences in which sometimes the information from one part of the sentence is unrelated to the other parts. Therefore, the use of local information rather than the whole sentence could improve precision. To address it, the author proposes to use a CNN-CRF model called GRAM-CNN, which is served with biomedical word embeddings, character embeddings obtained by a CNN-based model and a POS tagging embedding to help the model to use the local information. The resultant vector is used in a broad CNN model which uses different kernel sizes to take into account different levels of local information. Following that, some scores are obtained from the CNN with a fully-connected network which feeds a CRF layer which models the entity tags.

Giorgi and Bader [53] uses Silver Standard Corpora (i.e., non-manually annotated corpora) in training to initialize a BiLSTM-CRF model. The objective of this pretraining is to apply a transfer learning process from the knowledge learned on a 'source' dataset to perform a task on a 'target' dataset by using the learned

parameters from the pretraining. This allows to improve the generalization of the model and reduce the amount of labeled data needed to obtain high performance. Therefore, it could be possible to reduce the amount of Gold Standard Corpora needed in training, which involves a very costly and time-consuming process making that these are usually not very large. It concludes stating that the results significantly improve in models whose target dataset has a low number of annotated entities and it barely affects on large GSC datasets (≥ 6000 annotated instances).

Dang et al. [44] explains the importance of the linguistic information that a BiLSTM model receives in problems such as the ambiguous use of abbreviations and its correct classification. For that purpose, a more complex embedding is obtained to feed the model, incorporating POS embeddings and abbreviation embedding apart from the highly used word and character embeddings. It concludes that each of the embeddings helps the model to improve its performance, outperforming slightly similar models which do not take into account too much linguistic information without incorporating an abbreviation or POS embedding.

Zhao et al. [148] implements a multitask learning model to perform recognition and normalization jointly. The multitask proposed model uses a parallel strategy in the use of tasks to receive explicit feedback to model the mutual enhancement effects between tasks. To this end, a BiLSTM-CRF-CNN stacked model is used with character-level obtained with a CNN model and biomedical pretrained word embeddings [98]. It is concluded that the mutual exchange of information between tasks helps to improve the results of both tasks.

Wang et al. [137] extends Crichton et al. [43] idea of multitask learning models incorporating character-level embeddings to a BiLSTM-CRF model. Moreover, three different kinds of multitask learning models are compared, whose difference lies in the level in which the multitask is implemented, i.e., what parameters are shared. One model makes use of shared parameters at the character-level BiLSTM model, other at the word-level BiLSTM-CRF model, and a third with both ones. It is concluded that sharing the maximum number of parameters between target and source tasks, i.e., at a word and character-level, is the best performing option.

Yoon et al. [144] extends Multi-task Learning idea [137] [43] presenting ColaboNet, a network which is composed of multiple BiLSTM-CRF models in which each of them is an expert on just one entity class. These experts help each other sharing knowledge with all the other experts. This is achieved by training each model individually on an entity class and then further training on the outputs of other models trained on the other entity types. Therefore, models take turns in their role changing between expert and collaborator roles. As a consequence, each model is an expert in a domain and helps to improve other model performance, leveraging multidomain information from the other models.

Giorgi and Bader [54] explores different techniques to achieve BiLSTM-CRF model improvement: Variational dropout, Transfer Learning and Multi-task modelling. Out-of-Corpus tests demonstrate that BiLSTM-CRF models generalize poorly outside the corpus on which they were trained and proposes those three methods that improve substantially the generalization of the model. It is concluded that the best out-of-of-corpus performing model is obtained through

the combination of multitask learning and the regularization of recurrent layers via Variational Dropout.

In Table 2.4 a summary of the previous overviewed methods is shown. As it can be inferred, improvements along this kind of models come through the way these models are used and the information which is served as input.

Year	Reference	Input	Model
2017	Habibi et al. [64]	BiLSTM Character + Word Embeddings	BiLSTM - CRF
2017	Gridach [59]	BiLSTM Character + Word Embeddings	BiLSTM - CRF
2017	Crichton et al. [43]	Word Embeddings	CNN
2017	Lyu et al. [93]	Att-based Character + Word Embeddings	BiLSTM - CRF
2017	Unanue et al. [130]	Handcrafted Features + BiLSTM Character + Word Embeddings	BiLSTM - CRF
2018	Luo et al. [92]	Handcrafted Features + BiLSTM Character + Word Embeddings	Att-BiLSTM-CRF
2018	Ju et al. [69]	BiLSTM Character + Word Embeddings	Stacked BiLSTM - CRF layers
2018	Zhu et al. [150]	POS embedding + CNN Character + Word Embeddings	Multi-kernel size CNNs - CRF
2018	Giorgi and Bader [53]	Character + Word Embeddings	BiLSTM - CRF
2018	Dang et al. [44]	Abbreviation + POS + Character + Word Embeddings	BiLSTM - CRF
2019	Zhao et al. [148]	CNN Character + Word Embeddings	Bi-LSTM-CRF-CNN
2019	Wang et al. [137]	BiLSTM Character + Word Embeddings	Multi-task Learning BiLSTM - CRF
2019	Yoon et al. [144]	CNN Character + Word Embeddings	Multi-task Learning BiLSTM - CRF
2020	Giorgi and Bader [54]	BiLSTM Character + Word Embeddings	Multi-task Learning + Variational Dropout + Transfer Learning BiLSTM - CRF

Table 2.4: Non-contextualized embedding models taken into account in the review.

Contextualized Embeddings Models

These kinds of models make use of embeddings which take into account the context of a word to produce a different embedding depending on its neighbouring words. Therefore, the embeddings generated for the tokens in a text also capture the context, which in many circumstances could be crucial to achieve a correct classification of ambiguous words. In the former models, it was necessary to implement a model on top of the embeddings to try to catch this contextual information. Now, with contextualized embeddings, it is possible to encode this information without the necessity of using another complex model on top of it. By contrast, a simple classification layer is used on top of the embeddings in order to classify a token as a Named Entity and therefore, a fine-tuning process

(see Fig.2.8) has to be done to adjust the parameters for that task and learn the parameters of the classification layer. Consequently, these kinds of embeddings could be easily adapted to be used as a classifier, being very useful in NER, which can be treated as a sequence classification task.

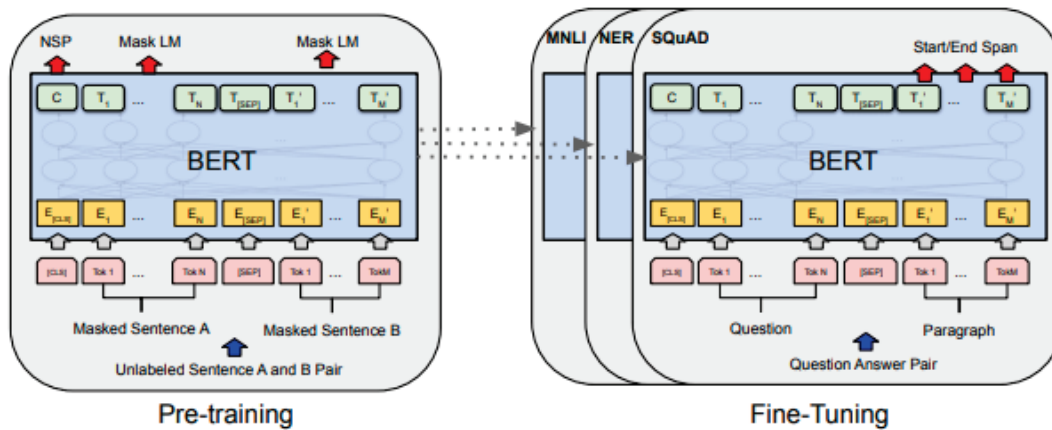


Figure 2.8: Fine-tuning process for multiple kind of tasks including NER. Image obtained from Devlin et al. [46]

The characteristics of these different kinds of embeddings, how do they work, and its differences were previously explained in section 2.2.3.2. In NER, the most widely used model has been BERT since it was firstly proposed in 2018 [46]. Its popularity mainly raises in the ease of adaptation to multiple tasks, which just requires minimal architectural modifications, its bidirectional encoding and its amazing results obtained in a great variability of tasks such as Question Answering, Relation Extraction and classification tasks such as Sentiment Analysis or NER among others. In great part, the success of a BERT model broadly depends on the training of that model. Originally, BERT was pretrained in general corpora: BookCorpus and Wikipedia. This allowed the model to perform well on general text, but in highly specific domains such as the biomedical domain, the accuracy of the model decreases significantly since the model can not handle the domain knowledge properly. Therefore, a solution to this problem is achieved by using an in-domain corpus to perform a pretraining allowing the model to incorporate that domain knowledge as part of the model [63] (See Fig. 2.4). This, as it was seen before in section 2.2.3.2, has been done in biomedical corpora obtaining BERT models [84] [61] [31] [24] [106] (see Fig. 2.2) highly adapted to the biomedical domain achieving state-of-the-art results in most of the aforementioned tasks, being BioNER one of them. Moreover, pretraining in the target task with unlabeled data or with data relevant to the task in which the model will be used has been demonstrated [63] to be beneficial in cases where the task data is a narrowly defined subset of a larger domain. Nevertheless, none publications were retrieved regarding this pretraining methodology for BioNER.

In the case of BERT, the architecture that these kinds of models adopt in NER is shown in Fig. 2.9. The publications retrieved in this review are the following ones, which correspond to the embeddings discussed in Table 2.2 adapted to a NER architecture as the one shown in Fig. 2.9.

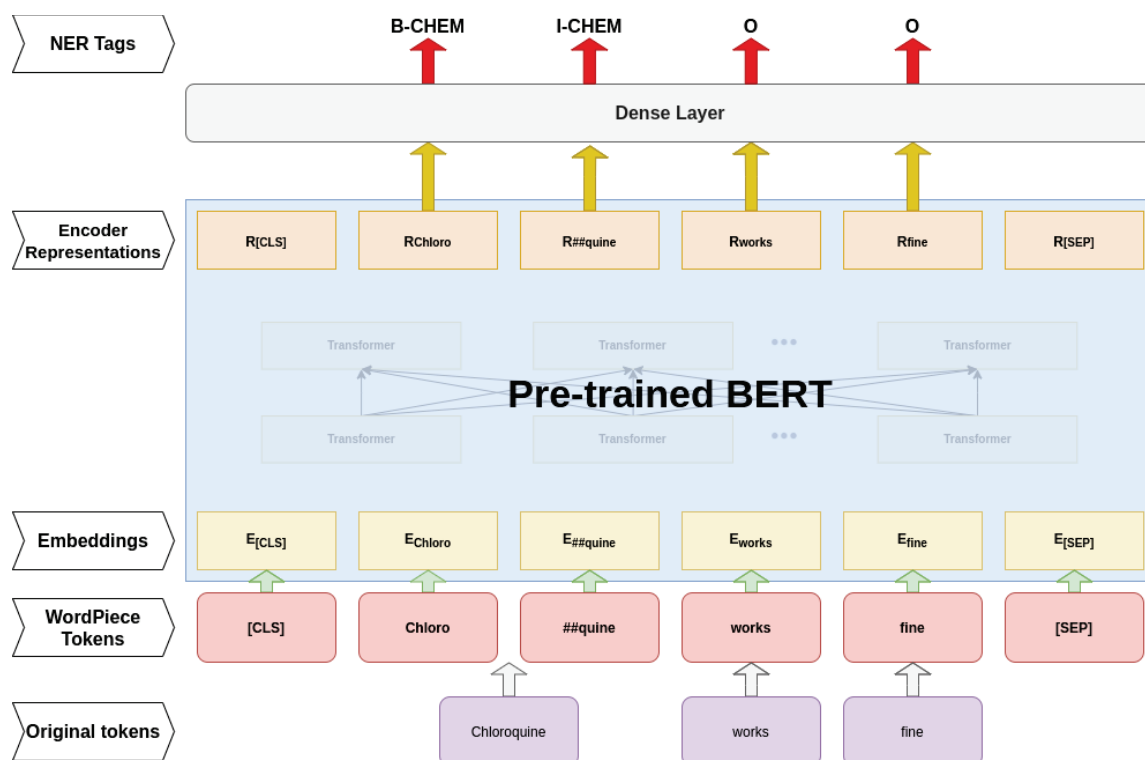


Figure 2.9: BERT architecture adapted to NER task

Lee et al. [84], which proposed BioBERT, achieved in addition to the improvement of results, facilitate the use and adaptation to different tasks with its model, making the use of this model one of the most widely adopted in BioNLP tasks such as BioNER since this publication was the one that had by far the greatest repercussion in terms of number of citations. This model can be easily fine-tuned in a downstream task such as BioNER, adapting that model to optimize its performance in the desired task with the use of a dataset for this task. This fine-tuning also involves minimal architectural modifications with the inclusion of a top layer adapted to the task we want to carry out, as seen in Fig. 2.9. In Lee et al. [84] BioNER task is tested along with other kinds of tasks resulting in state-of-the-art results in most of the dataset it was proven.

ClinicalBERT [24] offered state-of-the-art results in several tasks, including NER, in clinical domain texts. Nevertheless, since this domain differs slightly from the biomedical domain, later results [61] on tasks based on biomedical domain texts were proved to be lower than obtained by models directly pretrained on biomedical corpora. This highlights the importance of the specificity of the pretraining corpus used and how similar should be the task domain and pre-training domain to obtain better results possible.

Beltagy et al. [31], besides proposing SciBERT, performed several experiments in the implementation and training of this model concluding that the pre-trained and fine-tuned SciBERT model obtained almost similar NER results than BioBERT despite being trained on a substantially smaller biomedical corpus. Moreover, some experiments were done replacing fine-tuning with a frozen BERT embedding version where on top a BiLSTM-CRF layer was employed in the case of NER

as a task-specific architecture. Experimental results showed that performance was significantly worse than fine-tuning model and adding a simple classification layer on top. The objective of Beltagy et al. [31] with SciBERT model was to build a single language model useful across multiple science domains, not only the biomedical but also the computer science field.

Peng et al. [106], apart from proposing BlueBERT, introduces the Biomedical Language Understanding Evaluation (BLUE) benchmark to compare the performance of multiple biomedicine pretrained language models such as BERT or ELMO in a set of different kinds of tasks both on biomedical and clinical texts. The objective was to offer a framework in which to compare the overall performance of models and the robustness between tasks and domains.

Gu et al. [61], proposed a biomedical BERT model pre-trained from scratch (PubMedBERT) aiming to avoid non in-domain knowledge in the training of the model in order to try to specify the most the model in biomedical text. Apart from this model, Gu et al. [61] proposed BLURB²², a broad-coverage benchmark encompassing diverse biomedical tasks, previous attempts were GLUE [134] in general domain and BLUE [106] in biomedical domain, in order to supply to the community with a common framework in which diverse models can be compared and the progress can be tracked through thirteen publicly available datasets in six diverse tasks. In the leaderboard, the first position goes to PubMedBERT followed by BioBERT with a very similar global BLURB score.

Hybrid approaches

Every approach has its advantages and disadvantages, that is the reason why the combination of some of them could solve the weakness which other models may present. Most approaches combine the use of dictionaries with a Machine Learning model since Machine Learning approaches generally get better recall results and dictionaries have better precision results, improving consequently the F1-score with the hybrid model. Nevertheless, in more recent approaches like BERT, non-hybrid models were found mainly because of its main drawback which it is the computational cost which takes to train it. This is a type of disadvantage that cannot be solved by any other method.

The following ones are the publications retrieved in this review, which in some way make use of a combination of different kinds of models previously discussed:

Rocktäschel et al. [116] used a hybrid model with a merging result from two branches: one branch uses a CRF approach previously modelled (BANNER [80]) aiming to focus on morphological complex structures such as IUPAC naming conventions. The other branch uses ChemIDplus²³ dictionary converted to a deterministic finite-state automaton for getting linear time complexity in text matching. Following this matching, some post-processing rules are applied to try to correct the boundary detection. Finally, the results between the two branches are merged and the overlapping results are resolved following some rules in which CRF results are prioritized over a dictionary in terms of boundary detection in the overlapping.

Another hybrid model was the one proposed by Wei et al. [140] which make use of

²²<https://microsoft.github.io/BLURB/>

²³<https://chem.nlm.nih.gov/chemidplus/>

results from a Recurrent Neural Network and a CRF model separately in such a way that one branch was composed by a CRF model which used handcrafted features and another branch with a Bi-RNN model which used word embeddings as input. Results from both branches were merged, including the predicted labels from both models, a confidence score, and word embeddings from the penultimate layer of Bi-RNN model along with POS tags and list lookup features. All this is used as input to an SVM classifier that performs the entity tagging.

2.2.3.4 Post-processing

Once a NER model has been applied, a set of entity candidates is established. Some post-processing steps are usually employed to improve the recall and precision of the final result. This post-processing step can improve the quality and accuracy of the outputs by resolving the disambiguation of terms, parenthesis mismatching, and abbreviation ambiguities. It is worth mentioning that not always all NER systems require or use this step since most modern approaches such as the ones based on a biomedical pretrained BERT, do not necessarily need to make such an intensive use of post-processing steps since the results are generally higher than the former approaches which need to make use of several additional post-processing steps to obtain proper results.

As aforementioned, one of the main tasks usually performed is a boundary detection check in which some rules are usually employed to detect if left and/or right boundaries of entities may be displaced to obtain a more exact entity tagging resolving some abbreviation ambiguities. These rules are usually designed specifically for each implementation of a NER model since they are usually focused on the error analysis performed in that model and therefore must be designed jointly with the model.

Some models also employ co-referring solutions to address the inconsistency problem in which an entity is tagged in a certain part of the text but not in later parts or in which a certain entity is referred using multiple forms. This is usually tackled by using an attention mechanism as it was the case of Luo et al. [92] model which used this mechanism on top of a BiLSTM-CRF model and BERT-based models which make use of inner self-attention mechanisms. For this purpose, also more general models have been proposed as it is the case of Lee et al. [85] model which is an end-to-end coreference resolution model based on BiLSTM, which idea has also been later adapted to the biomedical domain [128].

Another often employed process is the resolution of abbreviation ambiguities. Because of the fact that in the biomedical field a given abbreviation can have multiple senses, its context is crucial and depending on it and the entity class it belongs, one of its meanings should be inferred. Multiple tools have been developed for this purpose as it is the case of Yu et al. [145], Schwartz and Hearst [120], Gaudan et al. [50], Sohn et al. [124] and Stevenson et al. [126] among others.

Normalization techniques are the most widely employed technique in BioNER pipelines since biomedical terms are usually polysemic and therefore multiple terms can be used to refer to a certain concept. Consequently, normalization techniques, which are often referred as Named Entity Normalization (NEN), are usually employed to map a recognized entity to a curated database which contains updated and curated data maintained over time by an organization and which it is widely adopted by experts

in a certain domain. Great part of these techniques are usually employed following the NER process and therefore the resulting NER entities are the terms used in NEN [139] [71] [140]. This can produce the propagation of cascading errors, since False Negatives are not considered in the NEN process and entities incorrectly classified could not be correctly normalized. This problem is usually tackled by performing NER and NEN jointly as some of the aforementioned NER methods implemented as it is the case of TaggerOne [81], tmChem [83] and Zhao et al. [148] model.

2.2.4 Evolution of methodologies

Along the review, it can be inferred the trend that it is observed in Fig.2.10. First attempts in making NER systems come in the form of hand-crafted rules, which sometimes make use of dictionaries to support the rule generation. These rule-based models used some features which were obtained through a feature engineering process focusing on different kinds of features in the text. Subsequently to this approach, Machine Learning methods prevailed over rules since it was not necessary to have to design the rules which handle the classification of entities, these methods establish patterns automatically through training with annotated corpus with entities. Following these advances, non-contextual embeddings come into play resulting in a distributed representation of words in a way in which we were able to capture different features in text automatically. These embeddings were used by different sorts of Recurrent Neural Networks (RNN) like LSTMs as the input, allowing RNNs to contribute with contextual information to these embeddings. A top classification layer like CRF was used in the output of these RNNs to perform the classification task of defining the entity type of each word. Last advances come in the form of deep contextualized word representation models like BERT, which are capable of giving a contextualized distributed word representation of a word in such a way that the same word has a different vector depending on the context where it appears. These kinds of models have been probed to work excellent as language models and for a task such as NER they just need to be fine-tuned adding a top-level classification layer which performs the final entity labelling in text. Therefore, it can be seen that apart from an increasing improvement in the results obtained in BioNER, the implementation of these models has become much easier by automating the way the featurization of texts and classification are carried out. In state-of-the-art methods, it is no longer required to have an intensive domain knowledge since the costs of designing patterns to capture within the domain are automatically inferred.

2.3 Datasets

The trend over the years of these models has shifted toward data-centric approaches. Former models use a set of features which had to be extracted manually as a part of an intensive feature engineering process. Following that, the patterns used for the classification of these handcrafted features also had to be manually designed, attending to the creation of rules which could capture the characteristics of an entity class. From the previous sections, it can be seen how this trend has shifted increasingly making use of large amounts of data. From this data, newer approaches aim to automatically infer the patterns underlying entity words without the necessity of having to manually design them. This patterns are obtained from the use of large amounts of data and therefore the trend has been to increase the availability of this

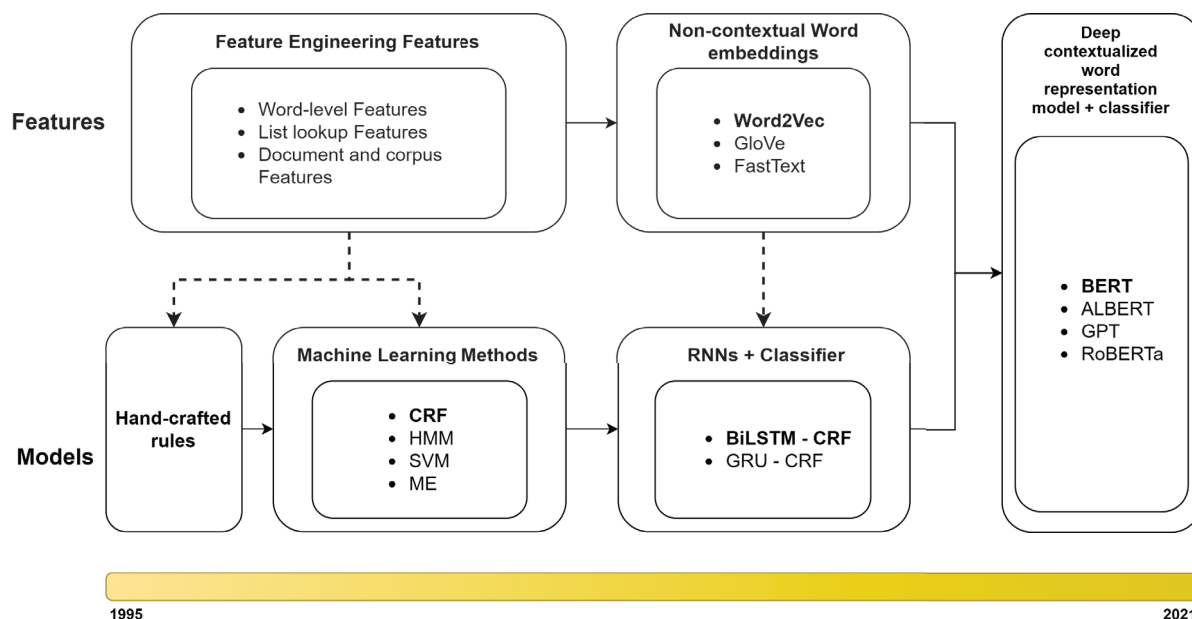


Figure 2.10: General trend observed in the implementation of NERs. In bold it is marked the model which was most widely used in each time.

data, thanks in great part to the shift towards digitalization which in recent years have been experimented.

Most recent models, which are based on embeddings, firstly use unsupervised learning to train these embedding models, which is the part in charge of representing the features underlying the tokens which form a text. Since unsupervised learning barely needs any preparation of the training data which is used, these models are usually trained in large collections of texts, articles, etc. . . as it could be Wikipedia, PubMed, MIMIC-III, etc. . . Therefore a massive training is usually done to capture a great amount of features in terms of linguistic value within these embeddings. Non-contextualized embeddings do not require as much computational power as Contextualized Embeddings due to the difference in the depth and complexity of its proper models, making that the appearing and developing of the former ones has only been possible in more recent years with the increasing amount of computational resources available with the apparition of high performance supercomputing GPUs and TPUs.

Once an embedding model has been trained, NER systems aim to use them as part of a system which aims to use their benefits to achieve a correct representation of the tokens which take part in a text. This representation is used in a NER model making a classification which aims to tag correctly tokens as entity classes. In recent approaches, the way this classification is done is learned from a vast amount of previous examples in which is exemplified how a token corresponds to one entity class or another. Therefore, these models need a previous training in which they learn the proper parameters which capture the patterns underlying entities to later make an appropriate classification of them. A corpus with annotations of the appearing entities is then needed to train these models and therefore these will be highly dependant on the quality of the annotations which were made on that corpus. These kinds of corpus are also used in the evaluation of the performance of the model to compare

the model with others and to understand the behaviour of the system based on those results and the error analysis which could be made on that model with the purpose of detecting which kinds of errors are systematically repeated.

In the literature it is common to attend to the following distinction in which depending on the quality of the annotated corpus, we have the following corpus:

- **Gold Standard Corpora (GSC):** the annotations within this corpus are hand-made by expert annotators in their domain. A guideline with the criteria on which they are based on to perform these annotations has to be made to make clear to subsequent users how is the granularity of these annotations and how specific these are. Due to the fact that all this process is done manually, it will be very costly and time-consuming, making that the effort to develop this kind of corpus is enormous and therefore not very large GSCs are usually found. In Table 2.5 a list of relevant GSCs in the biomedical domain is shown.
- **Silver Standard Corpora (SSC):** In this kind of corpus, the annotations are automatically done by existing state-of-the-art systems. The annotations achieved by multiple systems are then harmonized by establishing some rules in which a consensus is set for establishing or not a token as a certain entity class. Therefore, both time and effort in the annotation process is drastically reduced, making it possible to produce a very large corpus but with a lower quality in the annotations since some noise is present underlying these annotations.

The main part of this corpus is usually produced in the framework of annual challenges organized for the development of systems capable of carrying out a set of tasks in a given domain. This is the case of challenges like BioCreative²⁴ and BioNLP²⁵, among others, in which in some occasions BioNER tasks have been proposed offering a proper annotated corpus to carry out the challenge. In the following subsections, some of the more widely annotated entities in the biomedical domain are presented in addition to the datasets in which are captured. Moreover, some results from previously studied models are compared in different datasets.

2.3.1 Entities

In the former section, the importance of the quality of the annotations was brought to the forefront. The guidelines followed by annotators in GSC will determine how this annotations are and the classes in which the annotations are classified. These criteria is the ground truth on which subsequent data-centric NER models are based on to determine how a token is classified. An important criterium of this classification is how grained is this classification. An entity could be classified as multiple classes in which some of them are subclasses of other classes and then how general or specific is the distinction of these classes is an important criterium to take in mind in the design and use of corpus in the training of models. In relation to this granularity, the following distinction is usually done:

- **Coarse-grained:** More general criteria is used in terms of distinction between entities. Multiples subcategories and sub-classes are considered as the more general class they belong to. For instance, different kinds of diseases such

²⁴<https://biocreative.bioinformatics.udel.edu/>

²⁵<http://www.bionlp-st.org/>

as cardiovascular diseases or autoimmune diseases are categorized just as the entity class *Disease*.

- **Fine-grained:** A distinction is made between the sub-classes that comprise a class. How grained is this distinction depends on the guidelines followed in the annotation and on the desired objective in which the corpus is based on. For instance, a distinction between diseases could be done based on the standard classification performed on ICD-11²⁶ as it can be seen on Fig. 2.11

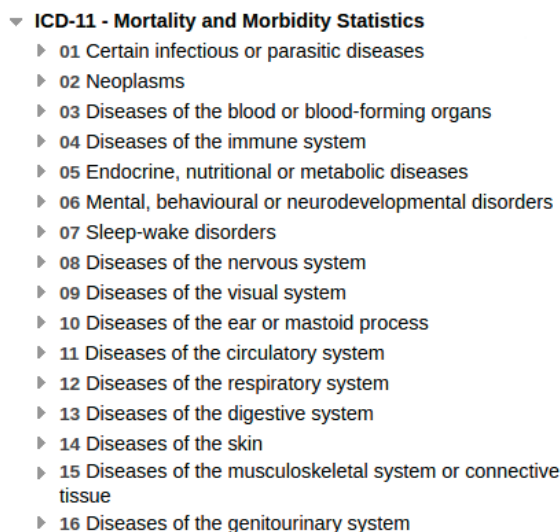
- 
- The image shows a screenshot of the ICD-11 Mortality and Morbidity Statistics classification. It is a hierarchical list starting with a dropdown arrow and the title 'ICD-11 - Mortality and Morbidity Statistics'. Below this, there are 16 numbered items, each preceded by a right-pointing triangle. The items represent different body systems and their associated diseases.
- ▼ ICD-11 - Mortality and Morbidity Statistics
 - ▶ 01 Certain infectious or parasitic diseases
 - ▶ 02 Neoplasms
 - ▶ 03 Diseases of the blood or blood-forming organs
 - ▶ 04 Diseases of the immune system
 - ▶ 05 Endocrine, nutritional or metabolic diseases
 - ▶ 06 Mental, behavioural or neurodevelopmental disorders
 - ▶ 07 Sleep-wake disorders
 - ▶ 08 Diseases of the nervous system
 - ▶ 09 Diseases of the visual system
 - ▶ 10 Diseases of the ear or mastoid process
 - ▶ 11 Diseases of the circulatory system
 - ▶ 12 Diseases of the respiratory system
 - ▶ 13 Diseases of the digestive system
 - ▶ 14 Diseases of the skin
 - ▶ 15 Diseases of the musculoskeletal system or connective tissue
 - ▶ 16 Diseases of the genitourinary system

Figure 2.11: Classification of diseases done in ICD-11. Extracted from <https://icd.who.int/browse11>

The challenges in fine-grained NER are the significant increase in NE classes and the complexity introduced by allowing a named entity to have multiple NE subclasses, which often presents overlapping situations. The distinction between subclasses could make it difficult to carry out the correct sub-categorization of entities, resulting in much lower performance on NER models. Consequently, the most widely used distinction is coarse-grained, followed by a normalization process in which we aim to achieve some of this subclass distinctions with the correct entity linking of a named entity to a curated concept in a controlled vocabulary thesaurus. These thesaurus are usually hierarchically or ontologically distributed and therefore subclasses and relations between these concepts could subsequently inferred.

The most widely adopted distinction between entities done in the biomedical domain is the following one:

- **Diseases:** Disease entities often have multiple naming variations. This further makes it more difficult to achieve a correct recognition and normalization of disease entities. In several articles, a more descriptive naming is often used for referring a certain disease. For example: lung infection with *Mycoplasma pneumoniae* could be a descriptive way of speaking about Bacterial Pneumonia. Moreover, abbreviations and identifiers of curated databases are usually employed for referring certain diseases.

²⁶<https://icd.who.int/browse11>

- **Chemicals:** The fact that there are numerous and extremely heterogeneous ways of identifying chemicals makes it difficult to find mentions of them in text. This includes trivial names (e.g. water), brands (e.g. Veklury®), systematic IUPAC names (e.g. 2,5,5-trimethyl-2-hexene), generic names (e.g. Benzenes), molecular formulas (e.g. CH_3), abbreviated forms (e.g. DMA for dimethylacetamide) and identifiers of curated databases such as CHEBI²⁷ (e.g. CHEBI:145994).
- **Genes/Proteins:** This entity class implies multiple levels of genetic entities such as genes, genetic variants, DNA or RNA mutations, proteins, etc. . . Jointly with their vast number of ways of expressing these multiple kinds of possible entities. Some examples of multiple possibilities are as follows: TNFRSF14 as a gene name, Gcg/Acg as codon change on a genetic variant, p.Gly195Val as an amino acid change in HGVS format, etc. . . The context in which these terms appear is crucial to identify the wide range of terms present in this entity class.
- **Species:** Organisms are usually classified following NCBI taxonomy²⁸. The ambiguity and polysemy within this entity class is lower than in other classes since the use of curated terms is more extensive.
- **Cell Lines:** This entity class applies to a defined population of cells. These are used as a way of dissecting the internal workings of tissues in a controlled environment and therefore they are common entities in cell biology research articles. For example, HEK293 or human embryonic kidney-derived epithelial cells.

These entities have been found in all GSCs which appear in Table 2.5. A little part of these corpus takes into account a more fine-grained classification and consequently, in order to establish the number of entities found in every GSC, these fine-grained entities have been classified as one of the previous entity classes if possible. An exception is done with anatomy entity class which is an unusual class which can not be classified as one of the previous classes.

2.3.2 Performance

Models exposed in section 2.2.3.3 are usually tested in one or several corpus of the Table 2.5. These results allow to compare the different models proposed and establish state-of-the-art models in a given time frame. In Tables [2.6 - 2.9] results for the retrieved models are compared grouping them by entity class. In order to make a fair comparison between methods, just a corpus in which 2 or more models reported results were considered. All results were reported with its F1-Score since it is a measure that takes into account both precision and recall. Since most publication performs a set of experiments with multiple variations of the proposed models, just the better results from these experimental results were taken in result Tables. In most corpus, testing is done following an exact matching criteria. In the case of BC2GM corpus, both exact matching criteria and alternative boundary matching criteria are offered and therefore, depending on the article, one or other results are shown. That is the reason why in Table 2.8 two columns are employed for each of the two criteria followed in BC2GM corpus. State-of-the-art results were marked in bold and the following best were underlined.

²⁷<https://www.ebi.ac.uk/chebi>

²⁸<https://www.ncbi.nlm.nih.gov/taxonomy>

State of the Art of Biomedical NERs

Year	Reference	Corpus Name	Entities	# Annotations	# Tokens
2014	Akhondi et al. [23]	BioSemantics	Chemicals	386110	5690518
2015	Krallinger et al. [74]	BC4CHEMD	Chemicals	79842	2235435
2004	Kim et al. [72]	JNLPBA	Genes/Proteins	35460	597333
			Cell Lines	4332	
2012	Bada et al. [27]	CRAFT	Chemicals	8137	560000
			Genes/Proteins	49961	
			Species	7449	
			Cell Lines	5760	
2008	Smith et al. [123]	BC2GM	Genes/Proteins	24583	508257
2010	Gerner et al. [52]	LINNAEUS	Species	4077	473148
2004	Kulick et al. [77]	PennBioIE	Genes/Proteins	17427	357313
2016	Li et al. [88]	BC5CDR	Diseases	12694	323281
			Chemicals	15411	
2016	Kaewphan et al. [70]	CLL	Cell Lines	341	6547
		Gellus		640	278910
2012	Pyysalo et al. [112]	BioNLP11EPI	Genes/Proteins	15811	253628
2014	Pyysalo and Ananiadou [110]	AnatEM	Anatomy	13000	250000
2013	Pafilis et al. [105]	Species-800	Species	3646	195197
2014	Doğan et al. [47]	NCBI Disease	Diseases	6881	174487
2013	Neves et al. [103]	Variome	Genes/Proteins	4309	172409
			Diseases	6025	
			Species	182	
2012	Pyysalo et al. [112]	BioNLP11ID	Genes/Proteins	6551	153153
2013	Ohta et al. [104]	BioNLP13CG	Species	21683	129878
			Anatomy		
			Genes/Proteins		
2013	Pyysalo et al. [113]	BioNLP13PC	Genes/Proteins	15901	108356
			Chemicals		
2010	Gurulingappa et al. [62]	SCAI	Diseases	2226	104015
2009	Leaman et al. [82]	Arizona - Disease	Diseases	3206	76489
2014	Bagewadi et al. [29]	mi-RNA	Genes/Proteins	1006	65998
			Species	726	
			Diseases	2123	
2012	Neves et al. [103]	CellFinder	Species	435	65031
			Cell Lines	350	
			Genes/Proteins	1340	
2013	Segura Bedmar et al. [121]	SemEval2013 - DrugBank	Chemicals	15745	≈ 65000
2020	Legrand et al. [86]	PGxCorpus	Diseases	635	≈ 35000
			Chemicals	1718	
			Genes/Proteins	1708	
2008	Pyysalo et al. [111]	BioInfer	Genes/Proteins	4162	33832
2015	Goldberg et al. [55]	LocText	Species	276	22550
2013	Segura Bedmar et al. [121]	SemEval2013 - Medline	Chemicals	2746	≈ 20000

Table 2.5: Details of the GSC found on biomedical domain ordered by their number of total tokens. These statistics were taken from Habibi et al. [64] or from their proper reference article

Almost all state-of-the-art results were obtained from the most recent models which are based on a pretrained version of BERT. BioBERT was the model which better results reported obtaining six out of eight state-of-the-art results and a second position in another corpus test. Just a bad result was reported in LINNAEUS [52] corpus were non-BERT models reported better results. The rest of the pretrained BERT models achieved results close to BioBERT but in most occasions one step below. Just in one occasion, the results from a non-BERT model surpassed BERT model which is the case of [53] in LINNAEUS corpus.

Due to the heterogeneity of the ways an entity can appear, it can be seen the importance of making use of contextualized solutions which can help to disambiguate terms based on their neighbouring tokens. This is one of the main reasons why this kind of solutions have become the state-of-the-art methodologies applied in NER tasks. Another reason why the design of NER systems is currently progressing in Contextualized Embedding model direction is because of its facility of adaptation to multiple tasks in which also state-of-the-art results are usually obtained [84]. Just a minimal architectural modification and a fine-tuning process is needed to adapt this models to a wide range of tasks.

Reference	NCBI - Disease	BC5CDR - Disease
Wei et al. [140]	-	84,28
Leaman and Lu [81]	82,9	82,6
Habibi et al. [64]	84,64	83,49
Crichton et al. [43]	80,37	80,46
Zhu et al. [150]	87,26	-
Giorgi and Bader [53]	84,72	82,32
Dang et al. [44]	84,41	84,68
Wang et al. [137]	86,14	**
Yoon et al. [144]	86,36	84,08
Zhao et al. [148]	87,43	**
Lee et al. [84]	89,71	87,15
Gu et al. [61]	87,82	85,62
Alsentzer et al. [24]	86,32*	83,04*
Beltagy et al. [31]	<u>88,57</u>	84,7*
Peng et al. [106]	88,04*	<u>86,6</u>

Table 2.6: F1 results in **Disease** entities in retrieved papers. In the case of ** in Wang et al. [137], only results from the overall score (88,78) with Disease and Chemical entities were reported. In the case of ** in Zhao et al. [148], only results from the overall score (87,63) with Disease and Chemical entities were reported. In the case of results reported with *, these were not taken from its own paper since they are not reported, they were taken from Gu et al. [61].

Reference	BC5CDR - chemicals	BC4CHEMD
Leaman et al. [83]	-	87,39
Leaman and Lu [81]	91,4	-
Habibi et al. [64]	91,05	86,54
Crichton et al. [43]	89,22	83,02
Luo et al. [92]	92,57	<u>91,14</u>
Giorgi and Bader [53]	91,64	-
Dang et al. [44]	93,14	-
Wang et al. [137]	**	89,37
Yoon et al. [144]	93,31	88,85
Lee et al. [84]	<u>93,47</u>	92,36
Gu et al. [61]	93,33	-
Alsentzer et al. [24]	90,8*	-
Beltagy et al. [31]	92,51*	-
Peng et al. [106]	93,5	-

Table 2.7: F1 results in **Chemical** entities in retrieved papers. In the case of ** in Wang et al. [137], only results from the overall score (88,78) with Disease and Chemical entities were reported. In the case of results reported with *, these were not taken from its own paper since they are not reported, they were taken from Gu et al. [61].

Reference	BC2GM (Exact)	BC2GM (Alternative)	JNLPBA - Genes
Campos et al. [36]	-	87,17	70,39**
Habibi et al. [64]	78,57	-	77,25
Gridach [59]	-	89,46	75,87
Crichton et al. [43]	73,17	84,41	69,73
Lyu et al. [93]	-	86,55	73,79
Zhu et al. [150]	-	87,26	-
Giorgi and Bader [53]	78,66	-	-
Wang et al. [137]	80,74	<u>89,06</u>	-
Yoon et al. [144]	79,73	-	78,58
Lee et al. [84]	84,72	-	77,59
Gu et al. [61]	<u>84,52</u>	-	80,06
Alsentzer et al. [24]	81,71*	-	78,59*
Beltagy et al. [31]	83,36*	-	77,28
Peng et al. [106]	81,87*	-	<u>78,68*</u>

Table 2.8: F1 results in **Gene/Protein** entities in retrieved papers. In the case of ** in Campos et al. [36] result is obtained from the average from 3 fine-grained entities: DNA, RNA and protein. In the case of results reported with *, these were not taken from its own paper since they are not reported, they were taken from Gu et al. [61].

Reference	LINNAEUS	Species-800
Habibi et al. [64]	<u>93,4</u>	72,1
Crichton et al. [43]	79,33	-
Giorgi and Bader [53]	93,54	<u>74,98</u>
Lee et al. [84]	89,81	75,31

Table 2.9: F1 results in **Species** entities in retrieved papers.

2.4 Source Code Availability

When you want to design a system based on the state-of-the-art models present at that time for a given task, it is important not only to know what models there are but also if the source code of these models comes along with the publication of that model. Not always the authors disclose the code implementation jointly with the publication, making it difficult to replicate the work they achieved. Some of these methods sometimes include a web-based demonstration to quickly check how the method works, and this aspect has also been studied in the next table. Another aspect which has also been evaluated is the license under which this source code is available. In Table 2.10, this availability aspects are studied jointly with links to the source code.

Reference	Source Code	License
Rocktäschel et al. [116]	https://github.com/rockt/ChemSpot	CPL-1.0
Zhang and Elhadad [146]	-	-
Campos et al. [36]	https://bioinformatics.ua.pt/software/gimli/	CC BY-NC-SA 3.0
Wei et al. [138]	https://www.ncbi.nlm.nih.gov/research/bionlp/Tools/tmvar/ ^a	U.S. Copyright Act*
Leaman et al. [83]	https://www.ncbi.nlm.nih.gov/research/bionlp/Tools/tmchem/ ^a	U.S. Copyright* Act
Wei et al. [140]	-	-
Leaman and Lu [81]	https://www.ncbi.nlm.nih.gov/research/bionlp/Tools/taggerone/ ^b	U.S. Copyright Act*
Habibi et al. [64]	https://github.com/glample/tagger ¹	Apache 2.0
Gridach [59]	-	-
Crichton et al. [43]	https://github.com/cambridgeltl/MTL-Bioinformatics-2016	MIT
Lyu et al. [93]	https://github.com/lvchen1989/BNER	GPL
Unanue et al. [130]	https://github.com/ijauregiCMCRC/healthNER	-
Luo et al. [92]	https://github.com/lingluodlut/Att-ChemdNER	Apache 2.0
Ju et al. [69]	https://github.com/meizhiju/layered-bilstm-crf	NaCTeM
Zhu et al. [150]	https://github.com/valdersoul/GRAM-CNN	-
Giorgi and Bader [53]	https://github.com/BaderLab/Transfer-Learning-BNER-Bioinformatics-2018/	-
Dang et al. [44]	https://github.com/aidantee/D3NER	-
Wang et al. [137]	https://github.com/yuzhimanhua/Multi-BioNER	Apache 2.0
Yoon et al. [144]	https://github.com/wonjininfo/CollaboNet	Equivalent to CC-BY
Zhao et al. [148]	https://github.com/SendongZhao/Multi-Task-Learning-for-MER-and-MEN	-
Giorgi and Bader [54]	https://github.com/BaderLab/Towards-reliable-BioNER	MIT
Lee et al. [84]	https://github.com/dmis-lab/biobert	Apache 2.0
Gu et al. [61]	https://huggingface.co/microsoft/BiomedNLP-PubMedBERT-base-uncased-abstract ²	MIT
Alsentzer et al. [24]	https://github.com/EmilyAlsentzer/clinicalBERT	MIT
Beltagy et al. [31]	https://github.com/allenai/scibert/	Apache 2.0
Peng et al. [106]	https://github.com/ncbi-nlp/bluebert	U.S. Copyright Act*

^a Demo: <https://www.ncbi.nlm.nih.gov/research/pubtator/>

^b Demo: <https://www.ncbi.nlm.nih.gov/CBBresearch/Lu/Demo/tmTools/demo/TaggerOne/demo.cgi>

¹ Neither trained model nor parameters is supplied. Just the general solution used.

² Model itself in Huggingface repository, non source code.

* License: <https://github.com/ncbi-nlp/bluebert/blob/master/LICENSE.txt>

Table 2.10: Availability aspects of the retrieved publications. In some cases none license was found, inferring then that it is a private tool.

Chapter 3

Biomedical Named Entity Recognition and Normalization System Implementation

A system for the recognition and normalization of biomedical entities has been proposed leveraging the previously made review knowledge. State-of-the-art solutions were studied and finally the system has been proposed based on this election, obtaining a system with state-of-the-art results on the recognition of entities. These entities have also been normalized as a final step based on the results obtained in the recognition step. In the following sections, details have been supplied about the system implementation and each of the elements which take part in this system. Finally, this system has been used for performing an annotation and normalization of the SARS-CoV-2 corpus CORD-19.

3.1 State-of-the-art method reutilization

Based on the previous BioNER State-of-the-the-Art methods, an election has been made to design a system for biomedical entity recognition and normalization. Some criteria have been established on which we have relied our election. These criteria attend to multiple characteristics of these methods and are as follows:

- **Availability:** Just methods with its source code available and open source licenses have been considered. Table 2.10 was consulted to look for this availability.
- **Results:** The performance of the model in BioNER task. In essence, what brings the most value to a NER system is how the selected model behaves in the detection of entities. Therefore, this was an essential criterium to take in mind, just state-of-the-art models were taken into account. Tables 2.6 - 2.9 were consulted for this criterion.
- **Ease of implementation:** Throughout the previous review, it has been seen how the appearance of NER models based on Machine Learning has supposed a shift on how NER systems are designed. This design changed from being handcrafted to being automatically inferred from vast amounts of data. This supposed a shift

in requirements making it easier to carry out implementations without extensive domain and language knowledge. This is another criterium which was taken in mind which state-of-the-art models broadly meet.

- **Ease of adaptation and reusability:** The previous change of focus also supposed that common model architectures could be reusable between different entities. State-of-the-art models also involved that with minimal model architectural adaptations, it could be leveraged in other NLP tasks. This is another criterium which was considered since a model with these characteristics may be easily extended in future works.

The model which better success on these criteria has been BioBERT [84] which is a BERT-based model pretrained in millions of PMC and PubMed biomedical articles. This model is open source under Apache 2.0 License, which means that it can be used and modified in further projects. It is the model which better results obtained in the considered datasets, resulting in 6 out of 8 state-of-the-art results on the corpus tested on Tables 2.6 - 2.9. As it is a model based on BERT, it could be easily implemented in a NER system since the model training, which is the hardest part in these models, was already done by the DMIS-Lab¹ team which is under the publication of BioBERT. To highlight the value of this, it is stated that it was necessary 8 NVIDIA V100 GPUs running during 23 days for the training. A costly challenge, difficult to achieve, that is only in the hands of not many organizations, laboratories, or universities. Finally, the adaptation and reusability criteria are broadly achieved since BERT-based models can be easily adapted to any task in which we could provide a proper dataset for a fine-tuning process in which this task is learned. Therefore, this model not only it is the state-of-the-art model but also it broadly satisfies the other established requirements.

3.2 Development

This project was proposed as an extension of LibrAiry bio-nlp tool² which offers a Web Platform³ and API⁴ solutions for the recognition of named entities. Initially, this tool was just proposed for chemical entities which were classified attending to the use of a SciSpacy [102] model jointly with a small set of morphological rules. The SciSpacy model used was trained on B5CDR corpus obtaining a F1-score of 84,49 which is far from state-of-the-art results which are around 90. A validation of the retrieved entities is then done through a normalization step aiming to improve False Positives rate for a precision enhancement. Nevertheless, since just a small database (6446 terms) for normalization was set, the number of False Negatives will also increase because this database does not cover all possible chemical entities detected and therefore some of them are not finally considered. Therefore, this validation improves precision at the expense of recall.

As it was established in the project objectives, an improvement of results is looked for. Moreover, an extension in the number of entity classes has also been done, extending this to disease entities and gene/protein entities. All this has been done based on

¹<https://dmis.korea.ac.kr/>

²<https://github.com/librairy/bio-nlp>

³<https://librairy.github.io/bio-nlp/>

⁴<http://librairy.linkeddata.es/bio-nlp>

Biomedical Named Entity Recognition and Normalization System Implementation

the previous state-of-the-art model selection, making that BioBERT has been the core piece in this BioNER system. Jointly with these NER improvements, the system has also been extended in BioNEN task by increasing the number of terms held in the proposed normalization database and extending the structure held for these terms for a larger information retrieved for each contained term. For these purposes, different processes had to be done along the project:

- **Fine-tuning:** This process has been done in BioBERT aiming to teach these models how to perform the task of tagging different entity classes. In order to address it as optimally as possible, a TPU was used. Therefore, the BioBERT implementation used has been adapted to this end through XLA Python package and all fine-tuning process has been held on Google Colab⁵ since this platform offers a free TPU within its Jupyter Notebook platform.
- **Model inferences:** Once the models have been fine-tuned, they have been used for inferencing entities from given texts. For this purpose, the implementation was designed for using a GPU if it is present on the machine where it is deployed.
- **Normalization Database:** A dockerized version of Solr has been used to ease the deployment of the database. Some scripts have been included to ease the reproducibility of this database along with all the terms which were processed in order to make use of them on this database for normalization. The collection of these terms have been done through the processing of a set of sources through Jupyter Notebooks.
- **Web Platform:** On the one side, back-end has been dockerized for easing its later deployment. On the other side, front-end has been deployed on GitHub pages.
- **CORD-19 Annotation:** A dockerized application has been deployed in an OEG-DIA server with 32 cores Intel Xeon (Cascade Lake) and 256 Gb RAM for the annotation of the entire corpus. The lack of GPU substantially slowed down this process.

3.2.1 Resources

3.2.1.1 Previous work reusability

Since this project arises as a continuation of an existent tool⁶, some ideas were reused along this project. This is the case of part of the Web Platform design. Moreover, the idea of performing a normalization through Solr have also been held but broadly extended.

3.2.1.2 Libraries and Services

The main programming language used for this system implementation has been Python 3.8 since it is the language where most model implementations are done. This is the case of BioBERT which is offered both as an extension of original BERT code⁷ and as a PyTorch-based BioBERT implementation⁸. The PyTorch implementation was the election since it makes it possible to easily incorporate further functions

⁵<https://research.google.com/colaboratory/>

⁶<https://github.com/librairy/bio-nlp>

⁷<https://github.com/dmis-lab/biobert>

⁸<https://github.com/dmis-lab/biobert-pytorch>

contained within PyTorch and Huggingface-transformers libraries. The following ones are the libraries used along the system implementation:

Python Libraries

- Pytorch [22]: open-source library for tensor calculations which are focused on deep learning models. It incorporates different kinds of neural network implementations which are useful for modeling different kinds of complex models like BERT.
- Huggingface-transformers [20]: provides different NLP model implementations jointly with NLP functions, which allows fine-tuning these models, inferring diverse task results, implementing them along a system... It also incorporates a repository⁹ where thousands of models are maintained being BioBERT¹⁰ one of these models.
- spaCy [19]: NLP framework which incorporates lots of text processing steps like POS Tagging, pre-processing steps, pattern matching rules...
- Flask [21]: Python web framework which allows to create web applications with the use of Python decorators.
- PySolr [18]: Python client which offers the interface of connection with an Apache Solr server.

Services - Solr

Open-source search engine based on Java and built on top of Apache Lucene, which is broadly used as an information retrieval library. It can be used for inverse index search, which is a way of structuring the information on search engines aiming to obtain results very fast. In order to achieve this, the fields of the docs (in this framework, terms along with other information: synonyms, ids... are called docs) within the database are previously indexed building an index in which the pre-processed tokens are contained jointly with the doc or docs in which they appear. Therefore, the following queries use this inverted index for quickly determining which is the doc which better fits the query. Scores obtained for determining which is the best retrieved doc are obtained through a cosine similarity Tf*idf based score.

This has been the component used for normalization since it allows us to perform multiple queries to multiple terms at a very short notice jointly with the possibility of containing multiple information for each of the terms since they are held as docs with multiple fields.

3.2.2 Workflow

In order to illustrate how the methodology has been applied, the Fig. 3.1 is proposed. The steps exposed in the top blue box are followed in each of the entity classes to build each of the components of the system. Based on the model election done (BioBERT), pre-processing steps were implemented to properly use the subsequent NER model, details are explained on Section 3.4.1. A fine-tuning process has been performed

⁹<https://huggingface.co/models>

¹⁰<https://huggingface.co/dmis-lab/biobert-v1.1>

using a pair of selected corpus for each class as it is later stated on Section 3.3.2, the discussion about the corpus selected for fine-tuning is also developed throughout this section. As a result, a BioNER model was obtained and on top of it a set of post-processing steps have been designed and implemented to improve the results obtained by the model itself, further details are given in Section 3.4.3. For the normalization process, a Solr database was populated for each entity class attending to a set of retrieved terms from relevant sources in each semantic type. Details about the process and the sources selected are given in Section 3.4.4. These steps finally built the system for the recognition and normalization of biomedical entities, which is the core part in subsequent platforms.

To show the performance of this system, two practical uses are also proposed. The first one is the use of the system in a web platform where a text to analyze can be given. Results from the processing are then shown in this web platform. Details are given in Section 3.5. Another use of the system is for annotating a widely used SARS-CoV-2 corpus: CORD-19 [135]. This system is the key part of an automated script which performed the annotation along the corpus. Further aspects are exposed in Section 3.6.

3.3 Creation of BioNER models

The development of a BioNER system was focused around the core piece of this system: the BioNER model. BioBERT has been the election to this end, being the part in charge of modelling the text span given and tagging the proper entity class for each of the words which make this text span up. For each of the entity classes in which this system has been focused on, a different fine-tuned model was implemented. Separated models were implemented (see Fig. 3.2) since experts models on each task were proven to behave with better results for its fine-tuned task than combining several entity classes in the same task obtaining just one model [63]. Results of the different BERT-based models taken into account in the review suggest that the more specific the model is, the better results will be obtained for this specific task. Separate models capture better patterns within each of the entity classes allowing to maximize its tagging performance, resulting in a system with the better model possible for each of the entities. Therefore, one model has been fine-tuned for disease entities, another for chemical entities and another for genes/proteins. These were the entity classes considered for the system, which are the most widely used classes in BioNER modelling and the ones with a high number of corpus available for a fine-tuning process (see Table 2.5 and Tables 2.6 - 2.9)

Apart from the BioNER models, the system has also been composed of other components such as post-processing procedures aiming to enhance the full system performance. An inverse index normalization in Solr is also part of the system. The architecture of this system can be observed in Fig. 3.2 and further details about each of the components are revised along the following sections.

3.3.1 Language representation

The value of transfer learning has proven to be important in the development of fields such as computer vision. Pretraining a complex neural network model with large amounts of data and then reusing it through a fine-tuning process have been

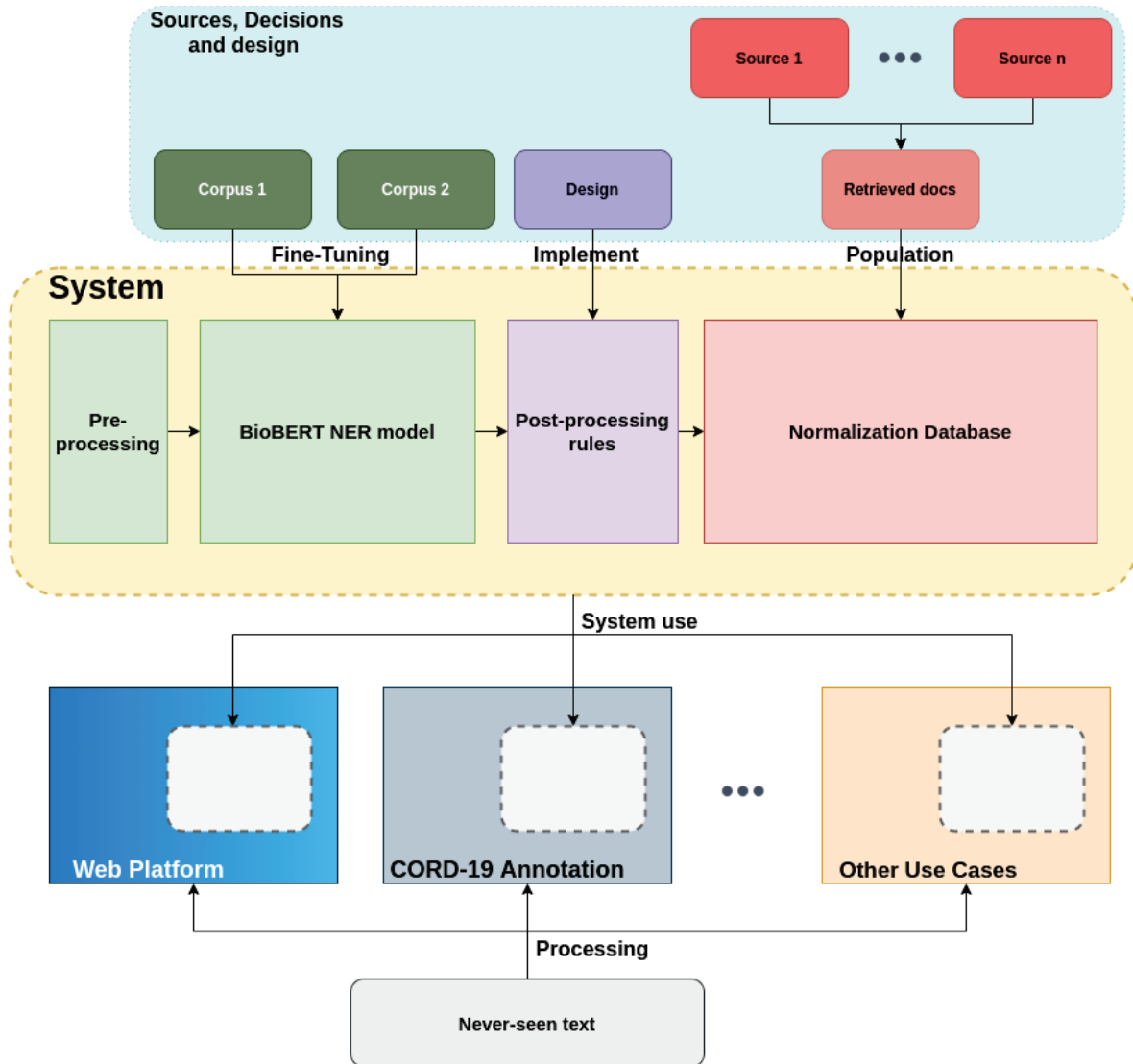


Figure 3.1: Overview of the methodology followed in implementation.

extensively shown to provide state-of-the-art results in a great part of tasks. In recent years, the appearance of models which make use of this approach in the NLP field has supposed the achievement of state-of-the-art results in most NLP tasks and domains. BERT models appeared as a response to the aforementioned. This model was possible thanks to a set of previous advances in NLP field which can be seen in Fig. 3.3 in which BERT is built on top.

Transformer [132], an attention mechanism that discovers contextual relationships between words in text, is used by BERT. Transformer contains two different functions in its original implementation: an encoder (see Fig. 3.4) that compiles captured information in a word to a vector and a decoder that generates predictions for a certain task based on the encodings. The main objective of BERT is to generate a language model and therefore just the encoder mechanism is necessary. In these encoders, two mechanisms are used: a self-attention mechanism, which is a technique used for capturing the most important contextual information for each word, and a feed-

Biomedical Named Entity Recognition and Normalization System Implementation

bio-nlp

<https://library.github.io/bio-nlp/>
<http://library.linkeddata.es/bio-nlp>

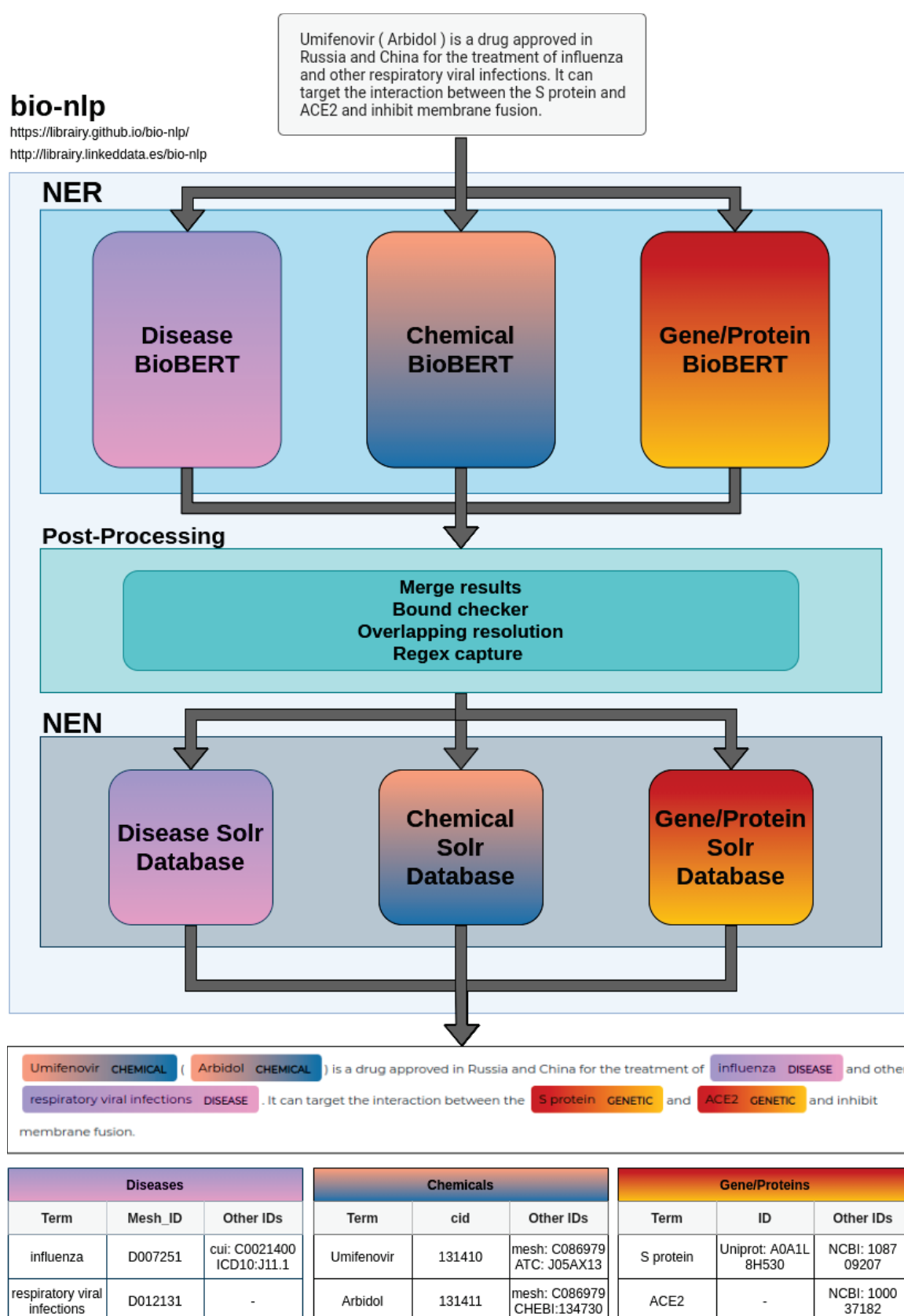


Figure 3.2: Overview of the system architecture.

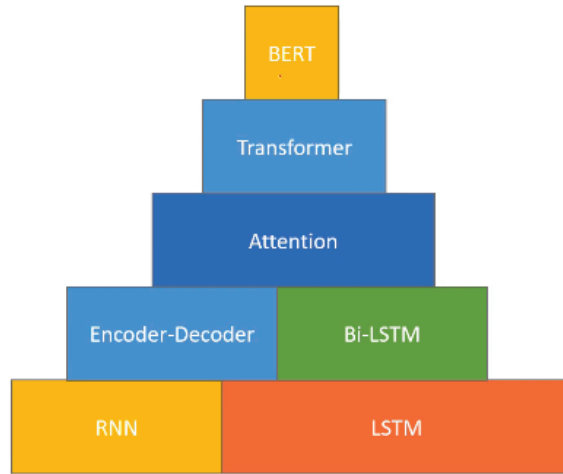


Figure 3.3: Mechanisms under BERT models are based on [12].

forward neural network. A stack of 12 Transformer-encoders is used in BERT_{BASE} architecture, a larger version of BERT (BERT_{LARGE}) was also developed making use of double number of encoders: 24. Thanks to the self-attention mechanism underlying these encoders, BERT is able to encode contextual information within a word. Moreover, data inputs are given bidirectionally, allowing it to encode contextual information for a word from all its surroundings, both left and right at the same time. Further details about its characteristics or the way training is performed were previously reviewed in section 2.2.3.2.

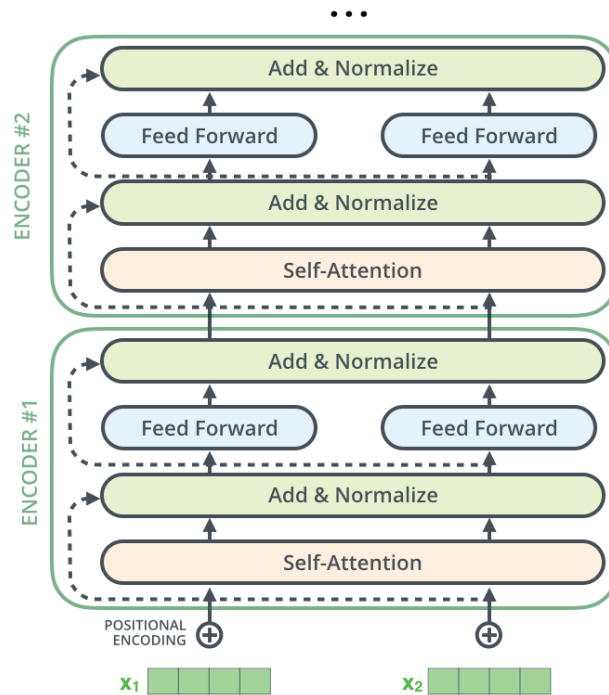


Figure 3.4: Transformer-encoders in which BERT is based on stacking them. Image adapted from [15].

BioBERT extends all these characteristics and through pretraining it on the biomedical domain, its parameters are adapted to a better behaviour on biomedical texts. The encodings which are offered for given words will represent better biomedical features which can be observed from the semantic properties which these present. As a means of illustrating some of these properties and how this model behaves on a set of biomedical terms, some visual representations have been made for its vectors. Since the dimensions in which these encodings are represented are high (768 dimensions), some high-dimensional visualization techniques have been applied, such as PCA and t-SNE. In Fig. 3.5, a bidimensional PCA vector space representation is shown. It can be seen how in general the different semantic types, i.e., the entity classes, are grouped. It is worth highlighting that protein and gene classes are generally mixed since they are high related semantic types. Actually, as it was seen in section 2.3, in BioNER both entity classes usually come together both in the datasets and the models developed. In Fig. 3.6 a three-dimensional t-SNE vector representation can be observed. The clusterings between entity classes which were stated in PCA are also seen in t-SNE. In fact, in t-SNE some sub-groupings begin to appear, meaning that in each of the entity classes there are also subclassifications which are also captured by BioBERT.



Figure 3.5: 2D PCA representation of 1024 biomedical terms

Even if we just take one entity class (for instance diseases) for representation, some clusters appear attending to some subcategorizations of these diseases. This is present on Fig. 3.7 where just disease entities were represented. In this representation, diseases are clustered attending to some subcategories in which diseases are divided. This is the case of some types of cancers which are jointly grouped, acid-base disorders, traumatology causes, etc. . .

The clusters found in these representations generally correspond to each of the se-

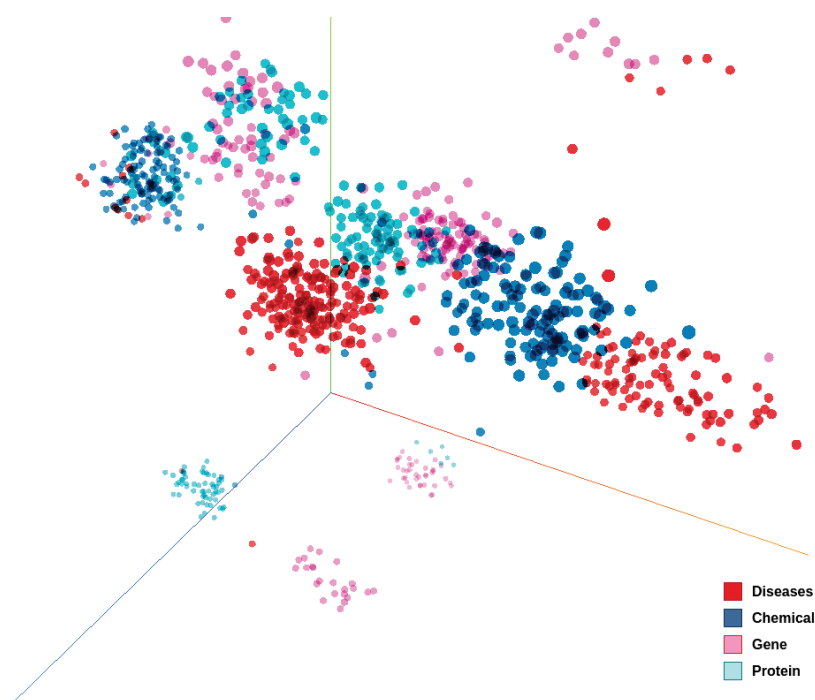


Figure 3.6: 3D t-SNE representation of 1024 biomedical terms

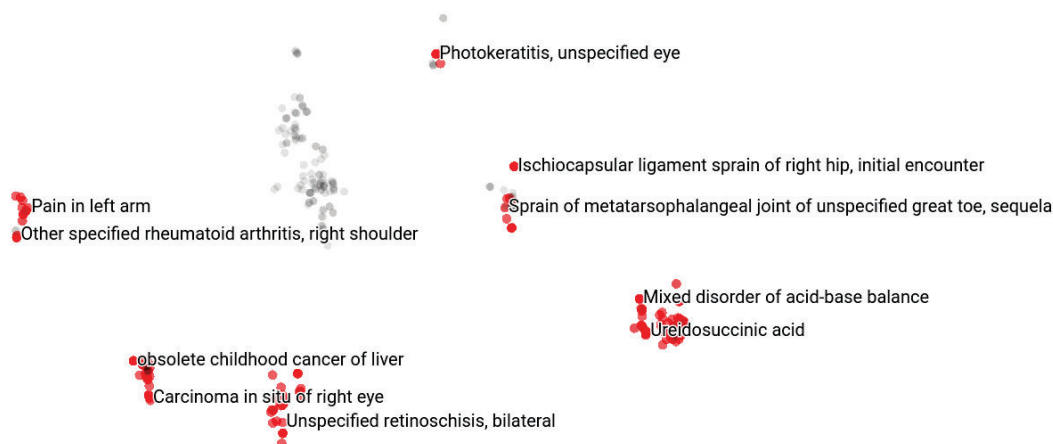


Figure 3.7: 2D t-SNE representation of a subset of disorders from previous represented biomedical terms.

mantic types which has been widely adopted in the literature (see Table 2.5) for establishing them as entity classes for BioNER tasks: Diseases, Chemicals and Gene/Proteins. Therefore, these representations help us to confirm the entity class division which we have adopted in the implementation of the system. A BioNER model for each of these classes was considered to achieve an state-of-the-art model for each entity class task. In this way, each model may better attend to the semantic properties implicit in the terms within an entity class.

3.3.2 Fine-Tuning

BERT-based models extract a high quality representation of a language and in the case of a in-domain pretrained model like BioBERT also of its proper domain. This can be used to extract precise language features from text data (see Figs. 3.5 - 3.7) or adapt these models to perform a certain task which in our case is a BioNER task for each of the target entities. The objective of this step is to automatically show the BioBERT model how to perform the task of tagging each of the entities through a fine-tuning process.

We have taken BioBERT model and on top of it an untrained fully-connected layer was added, as it is shown in Fig. 2.9). The reasons why this is the most widely used way of proceeding are as follows:

- Amount of resources: the hard part of the training was already done in the training and pretraining processes where bottom layers of our model are extensively trained. This required a lot of computational resources hardly affordable for most researchers. The resultant trained model, which encodes lots of information about our language and domain, has been leveraged to carry out a large range of tasks. Far less resources and time are needed for fine-tuning since much less intensive training is required with lots of less epochs.
- Amount of required data: Less amount of data is required since the highest intensive parts where the model learns language representations were already done. A much smaller corpus is needed for adapting the model to the required task. By fine-tuning the model, we can now achieve to train it to reliable results on a much reduced volume of training data.

Moreover, it has been proven that this fine-tuning procedure, with just one fully-connected layer on top and a few epochs training, achieves state-of-the-art results in a wide range of tasks as it was shown in Tables 2.6 - 2.9 with BioBERT.

Since the system is composed by three different models, at least three fine-tuning processes have been done. In the following section, a selection of corpus for fine-tuning is done.

3.3.2.1 Selected corpus

Since the way a task is learned broadly depends on the corpus where the fine-tuning process is done, the corpus which better adapts to the tasks we aim to do was selected. Moreover, we have also taken into account the corpus with the highest possible number of annotated entities. This can be observed in Table 2.5 where Gold Standard Corpus were collected jointly with the corpus size and number of annotated entities for each class.

In order to try to generalize the most the predictions done by the fine-tuned model, two corpus have been selected for each entity class for fine-tuning each model. This is done because the criteria underlying how entities are captured slightly differs between the corpus despite being the same target entity. These annotations are done in each corpus based on a given guideline in which curated annotators are based on. Using a pair of this corpus, we aim to supply the model with a better generalization power in situations where a never-seen text is passed. A discussion for each of the entity classes considered is held in the following paragraphs. It is worth mentioning

that the results obtained once fine-tuning has been performed were slightly worse than the ones given by the original paper [84], likely because of the hyperparameter search intensity and the number of epochs done is lower. In general, these results are around 1.5 F1-score points below.

Diseases

The two corpus in which more annotated disease entities were present are: BC5CDR - Diseases [88] with around 13000 annotations and NCBI - Diseases [47] with almost 7000. These corpus also correspond to the most widely used corpus in NER disease literature and most models provide results for each of this corpus (see Table 2.6), including BioBERT which obtained state-of-the-art for both tasks with an F1-score of 89,71 for NCBI - Diseases and 87,15 for B5CDR - Diseases. After our fine-tuning, these results were a few lower with and F1-score of 87,4 and 85,8, respectively.

Chemical

For chemical entities, the two selected corpus were BC4CHEMD [74] and BC5CDR - Chemicals [88] with around 80000 and 15000 entities respectively. The largest annotated corpus, BioSemantics [23], was not considered since the kind of text of which it is composed are patent texts which could slightly differ from biomedical articles which are the kind of texts in which the system has been focused on. Moreover, this corpus is not considered in the results of the reviewed models and therefore no comparison might be made for its performance. The selected corpus are also the most widely adopted corpus for NER tasks in chemical entities and most models provide performance results for these corpus (see Table 2.7). BioBERT obtained state-of-the-art results in BC4CHEMD with an F1-score of 92,36 and the second best result for BC5CDR with 93,47 which is almost the same than the state-of-the-art result obtained by BlueBERT [106] which was 93,5. Obtained F1 results after our fine-tuning were 92,99 for BC5CDR - Chemicals and 91,7 for BC4CHEMD.

Gene/Proteins

Gene and protein entities were jointly considered since they belong to very similar semantic types, as it could be seen in Fig. 3.5 and Fig. 3.6. Moreover, this consideration is also widely adopted in BioNER since most existent corpus consider them together. The pair of selected corpus were JNLPBA [72] and BC2GM [123] which offer around 35000 and 25000 annotations respectively. These were selected before CRAFT corpus [27], the largest Gene/Protein NER corpus, since most models report results based on those corpus and a comparison between them can be established. BioBERT reported state-of-the-art results on BC2GM results with a F1 of 84,72 and in JNLPBA results (77,59) were slightly worse than state-of-the-art which were reported by PubMedBERT with a F1 of 80,06. Results from our fine-tuning were a bit worse with 83,0 and 76,0 for BC2GM and JNLPBA respectively. Results on this joint entity class are significantly worse than other entity classes, perhaps due to the broad range of subentities classes which take part within this class. This makes that the amount of linguistic variability is enormous and harder to capture than the former entity classes.

3.4 Implementation of BioNER/BioNEN system

Once the models have been fine-tuned, they must take part as a core piece in the system. Rest of the parts of which the system is composed must be also developed and adapted. These have been summarized in Fig. 3.2. Both before and after the use of the models in NER task, some extra steps need to be done. The steps in which a NER pipeline is usually built were shown in Fig. 2.2.3 and detailed in former section 2.

Source code is available at: <https://github.com/librairy/bio-ner> jointly with documentation for its use.

Fine-tuned models used in the system can be found on Huggingface repositories:

- Diseases: https://huggingface.co/alvaroalon2/biobert_diseases_ner
- Chemicals: https://huggingface.co/alvaroalon2/biobert_chemical_ner
- Genetics: https://huggingface.co/alvaroalon2/biobert_genetic_ner

It is worth highlighting the impact which the chemical model achieved in its first month online in the Huggingface repository were it accomplished 76k downloads in just one month turning into the top 3 most downloaded model¹¹ in the token classification task and the first one related to the biomedical field.

In the following sections, details about the implementation of our system are shown.

3.4.1 Pre-processing

The first step which is needed in the system was a pre-processing in which we adapt the format of our text to serve as input to each of the three BioNER models. BERT-based models present a clear limitation in its input, the maximum length of tokens which can be passed is 512 because of a quadratic scaling limitation on self-attention blocks in which BERT is built. To address this issue, the processing of the given text has been done paragraph by paragraph and therefore the first step done in pre-processing has been to split a given text into the paragraphs in which are comprised. This implies that each model has to be used as many times as paragraphs we have, so the computing time will significantly rise. This is a drawback which we had to assume in the system to be able to face long text sequences such as biomedical articles.

Once the text has been split in its paragraphs of less than 512 tokens, we had to prepare each of them for model consumption. A WordPiece [118] tokenization step has to be made since this is the kind of token which BERT-based models use. BioBERT's vocabulary in which this tokenization is made is based on the original BERT [46] model's vocabulary which was built in its training process on Wikipedia and Book-Corpus data and therefore this vocabulary contains the most common characters, symbols, punctuation marks, n-grams and words found in training. This was studied in Table 2.2 in which it was observed that just BERT-based models which were trained from scratch make use of a vocabulary built from biomedical texts. Nevertheless, since BioBERT results were state-of-the-art, this seems not to lack the performance of the model.

¹¹https://huggingface.co/models?pipeline_tag=token-classification

3.4. Implementation of BioNER/BioNEN system

Following tokenization, special tokens are added to signal parts of the text, such as the first position with [CLS] mark and the end of the text with [SEP] mark. This tokenization is followed by a substitution of each of the tokens for an id which corresponds to each token in which we encode them for algorithm consumption. In Fig. 3.8 all pre-processing steps done are summarized.

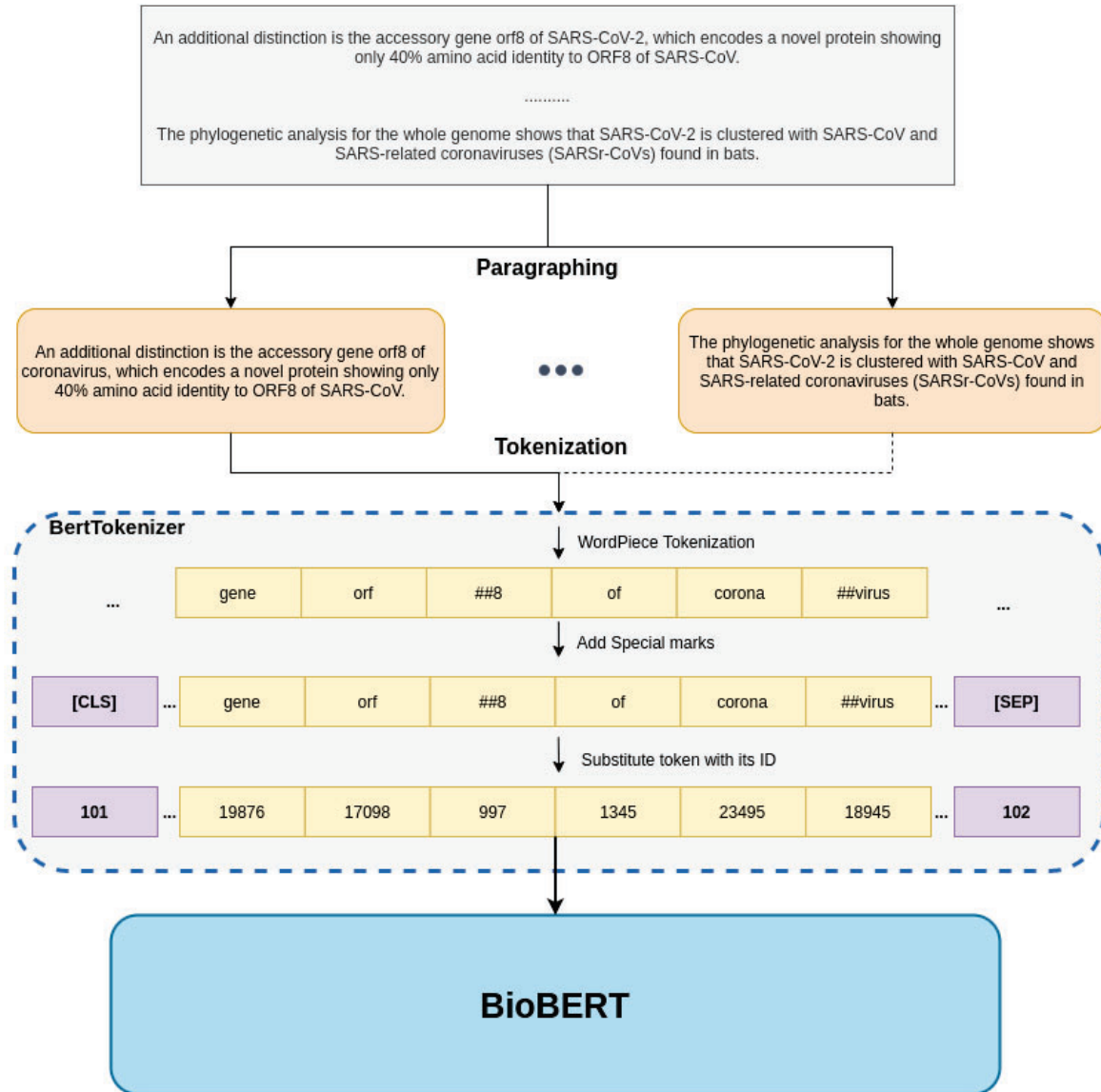


Figure 3.8: Pre-processing steps followed by the proposed system.

3.4.2 BioNER Modelling

The pre-processing step aims to prepare a text to serve as input to each of the NER models. Therefore, the pre-processed data is used in each of the proposed BioNER models, which takes care of producing the proper entity annotations within the given text for each of the proposed entity classes. The way BioBERT achieves this sequence

classification task has been previously described. A BIO tagging has been the option chosen for the annotation of entities. In this tagging, B stands for Beginning of an entity, I for Inside entity, and O for nonentities. As a result of applying each of these models, we will have a set of candidates in each of the entity classes. Some post-processing steps have been applied to these candidates to try to enhance the results and finally we have linked them to curated terms in a collection of curated vocabularies.

3.4.3 Post-processing

In order to refine and normalize results, a post-processing step has been done on top of BioBERT models. Once text has passed through the modelling step, a set of entity candidates is established and to improve the results of the entities on the retrieved set, some post-processing steps are applied. Moreover, a set of regular expression rules have been used to capture new entities which have been systematically observed to be ignored by BioBERT models. In the following sections, further details are given for each of the post-processing steps implemented.

3.4.3.1 Boundaries correction and Overlapping resolution

Some retrieved entities were just a part of a word. This is produced due to the fact that, in the BIO tagging which were produced as the output of BioNER models, some tags inside a word were incorrect producing that just a part of a word is tagged. In other occasions, some words are divided into subwords also due to incorrect BIO taggings. In order to try to correct this, a boundary correction step is applied before NER modelling aiming to extend properly the boundaries of the entities to solve the former problems. For example, if we had the disease "*influenza infection*" and it is annotated as follows: *influ* -> B-disease, *enza* -> O, *infection* -> I-disease. This post-processing step would join together "*influenza disease*" despite the fact that "*enza*" has been incorrectly tagged.

Overlapping entities were resolved using the larger entity in this nested entities since generally specifies better the entity referred. This is, on one side because the larger entity is the most complete entity to which an author refers to a certain entity and on the other side because the used framework, Spacy, does not originally allow nested annotations for entities. For example in "*p53 mutagenic cancer*", although the gene "*p53*" is present, the most complete entity would be the entire text span "*p53 mutagenic cancer*" with which we are referring to a disease caused by that gene. Therefore, we would just consider the larger entity which is the disease.

3.4.3.2 Capturing new entities with Regular Expressions

It was observed that COVID-related terms were not always correctly captured. Since one of the main objectives of the system was to carry out a COVID-19 corpus annotation, it was essential to establish all possible references to COVID and therefore a set of Regular Expression rules have been established to overcome the problem.

For entities which are similar to **SARS - CoV 2** the following regular expression is proposed. It also aims to capture MERS disease, a similar disease which publications are often referred to. Since sometimes this term is referred with SARS, another times with SARS infection and other possible combinations, the regex expression

3.4. Implementation of BioNER/BioNEN system

takes in mind these possible combinations to try to capture the maximum number of polysemies.

```
1 [/(sarsr?|mers)(\s?\-?\s?(covs?))?( \s?\-?\s?2)?(\s?\binfe.{1,10}?\b)?/gi]
```

For entities similar to **COVID - 19**, the following expression has been used, allowing some variability within the term.

```
1 [/(covid)(\s?\-?\s?(19))?( \s?\binfe.{1,10}?\b)?/gi]
```

Moreover, for terms similar to **coronavirus** the following expression has been established allowing also variability under which the entity could appear in text.

```
1 [/(coronavir.{0,6}?\b)(\s?\bpneumo.{0,8}?\b)?(\s?\binfe.{1,10}?\b)?(\s?\bdiseas.{1,6}?\b)?(\s?\-?\s?(20)?(19))?)?/gi]
```

It is worth highlighting that SARS-CoV-2 term and its related terms are ambiguous and could be used for referring also the specie of virus which causes the disease. However, since the use of this concept also involves the production of the infectious disease, the former captured entities will be classified as *Disease* entities.

The increasing number of variants of COVID-19 and its huge importance has boosted the number of literature related to this issue. Therefore, the following expression has been established for capturing **COVID-19 variant lineages**¹² which is the most widely used way of referring to COVID-19 variants.

```
1 [/( \b[A-Z]{1}\.\d{1,4}(\.\d{1,4}){0,4}\b)/gi]
```

Some chemical entities were observed to be ignored by the proposed NER model. Since some of them are easily recognizable through the presence of certain affixes, a regular expression was also set to ensure the capture of these kinds of terms in the case of NER models did not capture them.

```
1 [(\b\S{0,10}umab\S{0,10}?\b)
2 (\b\S{0,10}(feron|floxacin|zepam|prazole|triptyline|vudine)\b)]
```

3.4.4 Normalization

The way the normalization process has been addressed is through an inverted index search which is carried out through Apache Solr. One Solr core¹³ has been developed for each entity class to carry out further queries just on the belonging entity class. An extra core has been used for COVID-related drug target terms and proteins. This way, indexes can be built separately, resulting in an index and configuration for each entity class. Each core had to be populated with curated terms and related info which helps to map concepts to success with further queries. Therefore, a collection of terms and identifiers had to be developed for each entity class. This has been done attending to multiple sources to try to enlarge the number of entities within

¹²<https://cov-lineages.org/>

¹³Running instance which contains a single index and associated configuration.

Solr to maximize the chances of finding the queried entity candidate within our built database. A set of sources has been taken into account in each of the entity classes, mapping some of the concepts and terms between sources as it is described in the following sections. Most sources, with the exception of CTD [13] and PubChem [10], come from terms retrieved on BioPortal¹⁴ ontologies.

An schema was established based on which the desired fields have been set in the sources: a term field which is essential since it is the main concept designated for each retrieved concept, a synonym field holds all possible synonyms present for a given term, a type field establishes the most specific semantic type possible retrieved for a term and id fields in which we have from most important IDs, such as MeSH or CUI, to other more specific database cross references. This is the schema that in general the following Solr databases have followed with some exceptions in some fields. Since in the search of the normalized entity some synonyms are also considered, a broad range of possibilities is present in the resultant retrieved terms. This will allow us to choose the one with the higher score between different searches where we assign higher weights to either terms or synonyms, and we also assign different criteria where we look for strict matches or similar matches. From all this search space, the result with the higher score is considered.

Diseases

Four different sources have been used for retrieving disease terms, some mappings were established whether it is possible between terms to maximize the number of term identifiers in different databases of each term:

- **MeSH - Diseases** [7]: Medical Subject Headings¹⁵ is a thesaurus with hierarchical and controlled vocabulary which is produced by the National Library of Medicine (NLM¹⁶). This thesaurus includes thousands of terms regarding to several semantic types with disease-related terms among them. BioPortal includes an ontology version of this thesaurus from which we have extracted disease-related terms attending to the UMLS Semantic Type each term belongs to.
- **CTD - Diseases**: CTD's MEDIC disease vocabulary is a modified subset of the "Diseases" branch of the NLM's MeSH, combined with genetic disorders from the Online Mendelian Inheritance in Man (OMIM¹⁷) database. These terms have been merged with the previous ones through an outer join on MeSH IDs.
- **DOID** [5]: The Human Disease Ontology [117] is a comprehensive knowledge base of inherited, developmental and acquired human diseases. It integrates terms from a wide range of medical vocabularies such as MeSH, SNOMED, NCI, OMIM. . . therefore it has been used to extend terms which were not previously captured by the other sources. The way this was done is through an outer join on MeSH IDs.
- **ICD10CM** [6]: The International Classification of Diseases is a hierarchical classification listed by the World Health Organization (WHO), in which are encoded a wide range of signs, symptoms, abnormal findings, causes of damage, dis-

¹⁴<http://bioportal.bioontology.org/>

¹⁵<https://www.nlm.nih.gov/mesh>

¹⁶<https://www.nlm.nih.gov/>

¹⁷<https://www.omim.org/>

3.4. Implementation of BioNER/BioNEN system

eases, and/or other disease-related terms. The ICD-10-CM is the 10th version of this classification with a Clinical Modification of the source. Since this classification is used in its proper BioPortal ontology, further mapping concepts are added, which is the case of Unified Medical Language System identifiers (CUIs). The way this source extends the previous sources is through this CUI since not MeSH IDs are included. For that purpose, an outer join on this id has been done.

Once all the previous sources have been processed and mapped, the results were sent to an specific core in Solr which uses a configuration specifically designed for the desired fields which builds the index for the given terms. A total of 126985 terms were retrieved for disease normalization with Solr. This index is available at: <http://library.linkeddata.es/solr/#/bioner-diseases/core-overview>

Chemicals

Five sources have been considered for chemical terms, some mappings between terms have been done to increase the number of term identifiers in different databases and synonyms of each term. With the selected sources, it is aimed to capture the wide range of possible chemical entities which this entity class can include. From more general term names to IUPAC names, brand names. . . Details about the sources are the following ones.

- **PubChem** [10]: PubChem is the world largest chemistry open database and it is maintained by the National Institute of Health (NIH). Different classifications are made between the available terms, among which MeSH hierarchy has been the one used for our database. Therefore, approximately 130000 terms are considered since considering all possible terms in PubChem would hinder subsequent uses due to the large amount present, 381 millions. With MeSH hierarchy, it is expected to have the most widely adopted chemical terms within all the collection.
- **ChEBI** [2]: It is a chemical database which is mainly focused on small chemical components of molecular entities and therefore it complements other types of terms considered in the rest of sources. Any biological or synthetical component present in biological organisms is aimed to be captured on this database. An outer join on InChIKey has been used for connecting these terms with the ones present in the previous source. InChIKey is a hashed key of InChI, an International Identifier for chemicals, which offers an IUPAC identifier for an standardized codification of chemicals.
- **MeSH - Chemicals**: MeSH also includes thousands of terms regarding to chemical-related terms. BioPortal ontology version has been used to extract chemical-related terms attending to the UMLS Semantic Type each term belongs to. Since PubChem already includes MeSH terms, this source has been just used to add MeSH IDs and extend information from the previous terms. Therefore, this source has been combined with the previous ones through checking if the term is found either on term field or on the synonyms list. If it is not found, it has been appended to chemical terms.
- **CTD - Chemicals**: It incorporates terms from multiple chemical sources and therefore it has been used for complementing previously existent processed

Biomedical Named Entity Recognition and Normalization System Implementation

terms. It also helps to extend the retrieved information about previously considered terms. Non previously found terms have been appended from this source.

- **ATC** [1]: It is a classification of pharmacological substances organized in therapeutic levels. The ontology version of BioPortal has been the source considered for ATC since it incorporates further information and relations with other terms. Information regarding ATC level and ATC code was added to the previously considered terms. If the term is not present, it has been appended.

Processed and mapped results have been used on a specific core of Solr for chemicals in which apart from ids, terms, synonyms and types, the ATC level is part of the schema whether it was retrieved. The index for the given chemical terms has been built from a total of 344238 terms retrieved for chemical normalization in Solr. This index is available at: <http://library.linkeddata.es/solr/#/bioner-drugs/core-overview>

Genetics

This entity class is composed by a broad semantic type since it includes both gene-related terms and proteins. They are close semantic types and even in some occasions the use of the same expressions is diffuse. This has led to a wide range of terms within this entity class in which four large and complementary sources have been considered for trying to cover the biggest amount of entity variability possible.

- **GO** [4]: The knowledgebase underlying the Gene Ontology [25] is the largest source for the functions of genes and therefore it has been used aiming to capture terms related to genetic mechanisms.
- **OGG** [8]: The Ontology of Genes and Genomes [66] collects genes and genomes of certain organisms such as humans, virus and bacteria. Mappings to multiple sources are found in the BioPortal ontology. The previous source has been appended to this since the captured terms of each source are complementary.
- **PR** [9]: The Protein Ontology [101] contains a wide range of protein-related entities along with relations between them. This source covers the protein part since it contains a large amount of terms. It has been appended to the other sources since it is not mainly waited to contain overlapping terms with the other sources.
- **CTD - Genes**: It contains a vocabulary retrieved from multiple sources with a great variety of genes in multiple species. It has been used to extend the gene terms which were not previously captured, appending non-retrieved genes.

Another core has been used for genetic entities, populating an index with just that kind of entities. Since the genetic field is one of the widest fields, the number of retrieved entities is substantially larger than the previous classes. A total of 946584 genetic terms have been used in this Solr core. This index is available at: <http://library.linkeddata.es/solr/#/bioner-genetic/core-overview>

COVID-19

Since a practical case in which the system has been used is a COVID-19 corpus annotation, an extra normalization was looked for to try to capture genetic biochemistry drug evidences and targets for SARS-CoV-2 along with proteins related to the

virus. For this purpose, the COVID Data Portal¹⁸ has been used since it incorporates a list of these kind of terms retrieved from Open Targets¹⁹ platform. In order to try to extend this COVID-related terms a little more, also some protein-related terms were incorporated from the SARS-CoV-2 proteins found in the Protein Ontology. A total of 1404 terms were finally retrieved for this purpose, which builds an additional core in Solr. This index is available at: <http://library.linkeddata.es/solr/#/bioner-covid/core-overview>

3.5 Deployment as Web Platform

A web platform has been developed to ease the use of the proposed system. The main objective of this platform will be to generate a visual representation of each of the retrieved entity classes besides providing tables with the retrieved and normalized terms jointly with the found identifiers in the normalization step. In Fig. 3.9 an overview of the web platform is shown.

Its use is very easy, a text area is presented to provide the text we want to process. Once this has been set, the analyze button must be clicked in order to send this data to the system for processing purposes. This communication between the webpage and the system has been done through AJAX calls.

The system can be easily used through the web platform which is online at: <https://library.github.io/bio-ner/>

The dockerized web platform can be found in its Docker Hub repository: https://hub.docker.com/r/alvaroalon2/webapp_bionlp

3.5.1 AJAX calls

The provided text is sent through an AJAX call to the system to serve it as its input. This text is then processed and the results are returned through another AJAX call to the web page. Three kinds of results are returned: a visual representation of the text along with the found entities surrounded by a coloured box (see Fig. 3.10), normalized entities in different tables (see Fig. 3.11), one for each of the biomedical entities considered plus an extra one for COVID - related terms, and results in JSON format to produce a way of easily copying the retrieved normalized entities along with all its retrieved function.

3.6 CORD-19 Annotation

CORD-19 [135] is a collection of related-COVID-19 articles. It is offered as a freely available dataset to provide to the research community a research dataset in which apply natural language processing and AI-related techniques to produce new insights about the disease to help to fight against it. These kind of techniques are mainly necessary to help professionals to keep up with the research advances and updates about the infectious disease. Global efforts are focused on fighting it and therefore the amount of knowledge daily produced is large. As a result, the vast amount of

¹⁸<https://www.covid19dataportal.org/biochemistry?db=opentargets>

¹⁹<https://platform.opentargets.org/>

Biomedical Named Entity Recognition and Normalization System Implementation

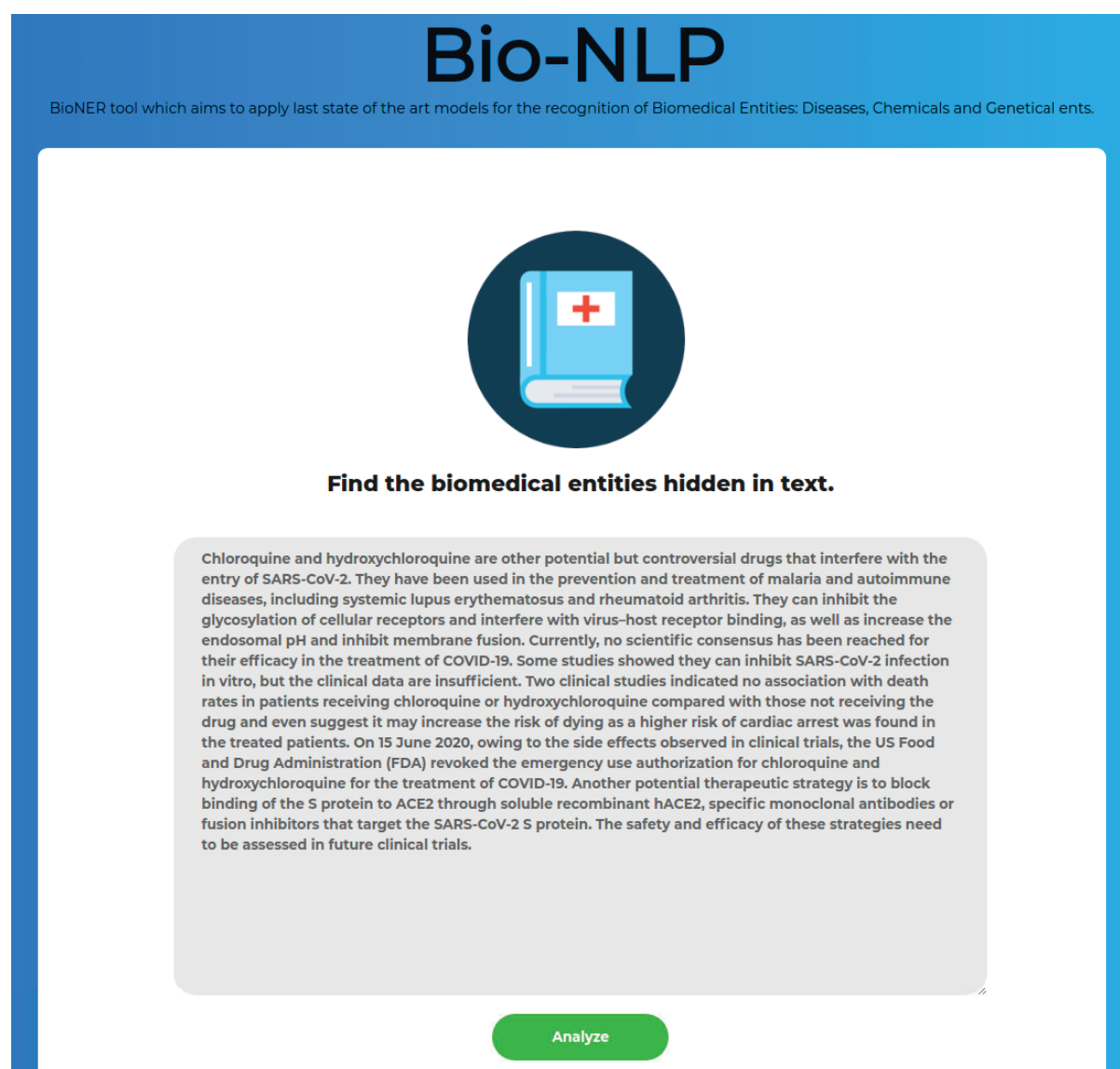


Figure 3.9: Overview of the proposed web-page for the use of the system.

scientific evidence accessible becomes a challenge in and of itself, particularly in instances where speedy choices are required.

Techniques such as the proposed, BioNER/BioNEN, help to face the problem underlying the vast amount of literature produced contributing to extract the embedded knowledge along the literature which is an information retrieval problem present in handling such amount of data quickly produced. As it was previously seen, NER and NEN are a key component in text processing tasks since the knowledge extracted will be subsequently used in other NLP-related tasks.

The proposed system has been used to extract the biomedical named entities: diseases, chemicals and genetic-related terms along with the proposed COVID-19 evidences along with its normalization (see section 3.4.4). This can be helpful in plat-

Results

Chloroquine CHEMICAL and hydroxychloroquine CHEMICAL are other potential but controversial drugs that interfere with the entry of SARS-CoV-2 DISEASE. They have been used in the prevention and treatment of malaria DISEASE and autoimmune diseases DISEASE, including systemic lupus erythematosus DISEASE and rheumatoid arthritis DISEASE. They can inhibit the glycosylation of cellular receptors and interfere with virus-host receptor binding, as well as increase the endosomal pH and inhibit membrane fusion. Currently, no scientific consensus has been reached for their efficacy in the treatment of COVID-19 DISEASE. Some studies showed they can inhibit SARS-CoV-2 infection DISEASE in vitro, but the clinical data are insufficient. Two clinical studies indicated no association with death rates in patients receiving chloroquine CHEMICAL or hydroxychloroquine CHEMICAL compared with those not receiving the drug and even suggest it may increase the risk of dying as a higher risk of cardiac arrest DISEASE was found in the treated patients. On 15 June 2020 DATE, owing to the side effects observed in clinical trials, the US Food and Drug Administration (FDA) revoked the emergency use authorization for chloroquine CHEMICAL and hydroxychloroquine CHEMICAL for the treatment of COVID-19 DISEASE. Another potential therapeutic strategy is to block binding of the S protein GENETIC to ACE2 GENETIC through soluble recombinant hACE2 GENETIC, specific monoclonal antibodies or fusion inhibitors that target the SARS-CoV-2 S protein GENETIC. The safety and efficacy of these strategies need to be assessed in future clinical trials.

Figure 3.10: Overview of the proposed web-page for the use of the system.

forms like Drugs4Covid²⁰ [28] which exploits the coronavirus literature to build an open catalogue of drugs jointly with a knowledge graph for its relations. In the workflow of this kind of technologies, an annotation of biological entities is often the first step based on which the subsequent steps are served. This is the case of Drugs4Covid which can use the retrieved entities for the population of a catalogue and a knowledge graph where an information retrieval task is held. The proposed system will be used for extending preexistent results on Drugs4Covid with retrieved entity classes terms and its normalization.

The way this annotation process has been handled is through the use of Solr, where the around 200000 CORD-19 articles²¹ were previously pre-processed in order to populate Solr with their paragraphs²² identifying the paper each of them belongs with an article id. Around 6,5 million paragraphs are obtained and held in Solr and therefore the proposed system will go through these paragraphs obtaining the entities along with its normalization.

The dockerized annotator can be found in its Docker Hub repository: https://hub.docker.com/r/alvaroalon2/bionlp_cord19_annotation

²⁰<https://drugs4covid.oeg.fi.upm.es/>

²¹<http://librairy.linkeddata.es/solr/#/cord19-papers/core-overview>

²²<http://librairy.linkeddata.es/solr/#/cord19-paragraphs/core-overview>

Normalized found terms

Diseases

Term	Found Term	Mesh ID	CUI	ICD10	Other Ids	Semantic type
SARS-CoV-2	Severe acute respiratory syndrome coronavirus 2	-	-	-	NCBITaxon: 2697049	-
malaria	Malaria, Vivax	D016780	C0024537	-	NCI: C34800, ICD10CM: B51, SNOMEDCT: Z7052006	Parasitic disease
autoimmune diseases	Autoimmune Diseases	D001327	C0004364	-	-	Immune system disease
systemic lupus erythematosus	Systemic lupus erythematosus, unspecified	-	C0024141, C2895176	M32.9	-	Disease or Syndrome
rheumatoid arthritis	Rheumatoid arthritis, unspecified	D001172	C0003873	M06.9	ICD10CM: M06.9, SNOMEDCT: 156471009, ICD9CM: 714.0	Connective tissue disease Immune system disease Musculoskeletal disease
COVID-19	COVID-19	D000086382	C5203670	U07.1	ICD10CM: U07.1, SNOMEDCT: 840539006, DOID: 0080600	Respiratory tract disease Viral disease
SARS-CoV-2 infection	post-acute COVID-19 syndrome	C000711409	-	-	-	Respiratory tract disease Viral disease
cardiac arrest	Cardiac arrest	D006323	C0018790	I46	ICD9CM: 427.5, SNOMEDCT: 30298009, ICD10CM: I46	Cardiovascular disease

Drugs/Chemicals

Term	Found Term	Mesh ID	CHEBI ID	ATC	ATC level
Chloroquine	Chloroquine	D002738	CHEBI:3638	P01BA01	5
hydroxychloroquine	Hydroxychloroquine	D006886	CHEBI:5801	P01BA02	5

Figure 3.11: Overview of the proposed web-page for the use of the system.

Chapter 4

Evaluation and Conclusions

In this section, it is assessed the performance of the system throughout different corpus. An error analysis based on the results of this system has also been done to illustrate what are the most common errors found to address them with further improvements. Some statistics have also been established about the annotation and normalization of CORD-19 corpus terms. Finally, some future work is explored along with the conclusions of the work.

4.1 NER Evaluation

In order to evaluate the proposed system for NER task, a pair of corpus have been used to test the obtained results against the underlying annotated ground truth found on these Gold Standard Corpus. The first one, PGxCorpus[86], has helped us to jointly assess the three considered entity classes. This corpus was selected from the ones present on Table 2.5 since is the only one in which we can jointly find the three considered entity classes. The second corpus, COVID-19 MLIA @Eval [17], has served us to evaluate how this system behaves in the presence of COVID related terms. In the following sections, details about these evaluations are given.

4.1.1 Datasets

4.1.1.1 PGxCorpus

PGxCorpus [86] is comprised by 945 sentences from 911 PubMed abstracts, annotated with pharmacogenomics entities of interest, which makes that we find Disease, Chemical and Genetic entities jointly with other nonstudied entities such as phenotypic related entities which in total are 10 entity classes. This corpus is also composed by relations between these entities which have not been used to test our system since the system it is not designed for this task. All of it is contained in Standoff format files [11]. For the preparation of this GSC, 11 annotators, out of which 5 were considered senior annotators, carried out the manual annotation with an inter-annotator agreement of 63,8 with an exact match criteria and 76,1 for partial match criteria in terms of Macro-Average F1-score. In Table 4.1 inter-agreement for the considered entities is shown.

The presence of several nested entities in these classes makes that two additional

Entity	Matching criteria	
	Exact	Partial
Chemical	76,8	82,1
Gene_or_protein	72,6	89,4
Disease	71	79,1
Macro Avg.	73,5	83,5

Table 4.1: Inter-agreement per entity type in terms of F1-score. Hierarchy was considered. Results extracted from Legrand et al. [86]

scenarios have been considered to avoid multiple annotations in the form of overlapping entities. The first scenario considers the largest entity found in a given overlapping, while the second scenario considers the shorter nested entity or entities which compose an overlapping. This has been done since the proposed system does not consider overlapping and therefore in some situations this would significantly lack system evaluation. An example which illustrates this can be seen in Fig. 4.1.1.1. Here the first thing to do is to select just the target classes: Disease, Chemical, Gene_or_protein. Afterwards, the two described scenarios are separately prepared. In the first one, we consider the largest entities in the overlappings and therefore in "combination of AZD7762 and olaparib", the labels for "AZD7762" and "olaparib" are dismissed, considering just the largest entity which is "combination of AZD7762 and olaparib". In the second scenario, we would consider the shorter entities: "AZD7762" and "olaparib" instead of "combination of AZD7762 and olaparib". The creation of this additional scenario also causes that in situations where different entity classes are overlapping, one of the classes is dismissed as it is the case of "p53 mutant pancreatic cancer" which in the case of the first scenario, is entirely considered as Disease dismissing "p53" as Gene_or_protein. In the second scenario the disease is dismissed but the Gene_or_protein "p53" is considered. All this has been done to assess the performance in the different scenarios since the proposed system does not take into account the presence of nested entities, i.e., it does not manage nesting entities. It considers the largest entity in the case of overlapping since it is usually the most complete term possible.



Figure 4.1: PGxCorpus example visualized by [125]

Moreover, the presence of discontinuous terms has not been considered since a coreference resolution step should be needed to try to face this kind of discontinuity between entities in which one part of the entity is continued in other part with a gap between them. This is something not covered by the proposed system.

4.1.1.2 COVID-19 MLIA @ Eval

As a means of evaluating COVID-19 related texts, the COVID-19 MLIA @ Eval manually annotated corpus [17] has been considered. This is the only manually annotated

currently found related to COVID-19 and it is a corpus developed as a community challenge in which the information extraction task consists of a NER task. Corpus was extracted from the Europe Media Monitor (EMM)/Medical Information System (MediSys), which is a collection of metadata automatically extracted from news articles related to Covid-19. This corpus is composed by sentences in which none relation is established between them and therefore they had to be processed separately in order not to influence the contextual information which BERT takes into account. From the available entity classes, we have just considered *sosy-dis* since it jointly considers signs, symptoms, and diseases. The rest of the classes does not fit the considered classes in our approach. The entity class *drug-trt* has not been considered since most of the annotated entities were observed to be "quarantine", "mental healthcare" and "isolation" which are terms far from the ones which our system aspires to annotate. Inter-agreement was not established since just an annotator performed the annotations, which means that the ground truth established in the corpus have to be carefully considered.

4.1.2 Evaluation Metrics

In most cases, NER systems are assessed by comparing their outputs to human annotations (GSCs), which are considered the ground truth underlying a text. Both exact-match and partial-match have been used to quantify the comparison [129]. Precision, recall and F1-Score metrics are computed on the number of true positives (TP), false negatives (FN) and false positives (FP) obtained in each matching criteria:

- True Positive (TP): entities predicted by the system and ground truth match.
- False Negative (FN): the proposed system does not annotate an entity present on the ground truth.
- False Positive (FP): an entity predicted by the proposed system does not match the ground truth.

Precision evaluates the capacity of the NER system to provide just accurate entities, whereas recall evaluates its capacity to identify all entities in corpus. F1-Score is the harmonic mean between precision and recall. These metrics were evaluated for each entity class and also across all of them.

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

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

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} = \frac{2 * TP}{2 * TP + FP + FN}$$

The macro-averaged metrics and micro-averaged metrics are two regularly used measurements for this purpose. The macro-averaged metrics calculate each metric separately for each entity class, then average them, and therefore all entity classes are treated equally. In the following example, this is exemplified with the precision metric but this is extended for recall and F1-Score.

$$Pr_{macro} = \frac{Pr_1 + Pr_2 + \dots + Pr_k}{k}$$

Micro-averaged metrics aggregate contributions from all entity classes from all classes to compute the average, making that all entities are treated equally. In the next equation the Micro-Average for precision is shown, this is extended for the rest of the metrics.

$$Pr_{micro} = \frac{TP_1 + TP_2 + \dots + TP_k}{(TP_1 + TP_2 + \dots + TP_k) + (FP_1 + FP_2 + \dots + FP_k)}$$

4.1.2.1 Exact-match Evaluation

True Positives will just be considered if the predicted left and right boundaries and entity class match with the ground truth. This makes that in situations where our system has not complete success retrieving boundaries but it has success identifying the entity, FP and FN will appear making that recall and precision will significantly lower.

4.1.2.2 Partial-match Evaluation

True Positives will allow that the left and/or right boundaries do not match, but the entity class is required to match between prediction and ground truth. This allows us to assess whether entity boundaries are the reason why exact match results worsen.

4.1.3 Experimental Results

The results of NER task have been given in all considered scenarios for PGxCorpus in Tables 4.2 - 4.4 both with exact-match and partial-match criteria. In the following Section 4.1.4.1, results are discussed.

	Larger Ents					
	Exact Match			Partial Match		
	Precision	Recall	F1	Precision	Recall	F1
Disease	0,59	0,62	0,61	0,78	0,81	0,80
Chemical	0,79	0,69	0,74	0,92	0,78	0,85
Genetic	0,56	0,61	0,58	0,84	0,91	0,87
Micro-Avg	0,65	0,64	0,65	0,86	0,84	0,85
Macro-Avg	0,65	0,64	0,64	0,85	0,83	0,84

Table 4.2: Results obtained in PGxCorpus considering a first scenario where larger entities where consider in case of overlapping.

In the case of COVID-19 MLIA @ Eval corpus, the results are shown in Table 4.5. In this case, none overlappings are found and therefore it has not been needed to consider different scenarios. A discussion about the results is held in Section 4.1.4.2

	Shorter Ents					
	Exact Match			Partial Match		
	Precision	Recall	F1	Precision	Recall	F1
Disease	0,54	0,73	0,62	0,65	0,84	0,73
Chemical	0,87	0,76	0,81	0,96	0,78	0,86
Genetic	0,62	0,62	0,62	0,91	0,90	0,91
Micro-Avg	0,71	0,69	0,7	0,88	0,84	0,86
Macro-Avg	0,68	0,70	0,68	0,84	0,84	0,83

Table 4.3: Results obtained in PGxCorpus considering a second scenario where shorter nested entities where consider in case of overlapping.

	Overlapped Ents					
	Exact Match			Partial Match		
	Precision	Recall	F1	Precision	Recall	F1
Disease	0,62	0,62	0,62	0,79	0,81	0,80
Chemical	0,91	0,67	0,77	0,96	0,75	0,84
Genetic	0,63	0,62	0,63	0,92	0,90	0,91
Micro-Avg	0,73	0,64	0,68	0,91	0,82	0,87
Macro-Avg	0,72	0,64	0,67	0,89	0,82	0,85

Table 4.4: Results obtained in PGxCorpus considering ground truth entities as given in the original corpus with the presence of overlappings with nested entities.

	Exact Match			Partial Match		
	Precision	Recall	F1	Precision	Recall	F1
Disease	0,53	0,81	0,64	0,59	0,90	0,71

Table 4.5: Results obtained in COVID-19 @ MLIA Eval considering the the only entity that our proposed is based on: Diseases

4.1.4 Discussion

4.1.4.1 PGxCorpus

Results change between the considered scenarios and match criteria and therefore a discussion about these variations is held.

Results obtained considering the scenario with the shorter entities (Table 4.3) were higher than in the scenario with the largest (Table 4.2) in the case of exact matches. This exposes that the system most of the time considers the nested entities which composes larger entities in cases like "*combination of AZD7762 and olaparib*" where we would annotate "*AZD7762*" and "*olaparib*" independently.

In the case of partial matches, the results are more similar, especially in the case of recall, since this partial match criteria makes that the captured entities along the text are low because False Negatives get substantially lower both in the first and second scenario. It is worth highlighting a larger improvement in precision results

with larger entities, this might be because in the scenario with shorter entities in occasions where a nested entity is of a different class from the larger, we would just consider the nested, dismissing the larger and considering a different entity class. This situation is almost always found in entities like "*EGFR -Mutant Lung Cancer*" (Disease) where the disease is the larger entity class and genetic or chemical nested are found inside: "*EGFR*" (Genetic). Since our system in these occasions would consider the larger entity found: "*EGFR -Mutant Lung Cancer*", then precision gets higher in larger entities scenario ("*EGFR -Mutant Lung Cancer*" is considered) and lower in shorter entities scenario ("*EGFR*" is considered).

Another case which is worth mentioning is that in the shorter entity scenario, the results both in recall and precision are equal in exact match and almost equal in partial match for genetic entities. This implies that $FP \approx FN$, which suggests that almost every entity which was incorrectly annotated was due to an incorrect boundary selection which in the case of exact matching entities like "*5-HTTLPR*" (Genetic) are tagged with incorrect class or boundaries ("*5-HTTLPR S allele*"). In the case of partial matching criteria, this is just because of the class. This entity detection misclassified makes that both FN and FP will jointly increase, resulting in this case in equal results on recall and precision since almost all errors are of this type.

In almost all provided results, recall is higher than precision or similar, the exception comes in chemical entities where precision is significantly higher than recall. This might be because chemical entities are worstly captured but better classified, which in other entities, the opposite is true. It was observed that more descriptive chemical entities like "*BRAF inhibitors*" (Chemical) were mostly not captured making that in that case, the nested entity "*BRAF*" (Genetic) is the predicted annotation. This suggests that maybe the fine-tuned BioBERT model for chemical entities has not been trained on this kind of descriptive entities. Moreover, this corpus is mainly focused on pharmacogenetics, making that the chemicals that appear are mainly highly specified in this subfield. Corpus for fine-tuning were more general, making that generalization in some occasions is not enough for identifying very specific entities.

Taking into account an scenario where all entities are jointly considered, i.e., with overlappings (Table 4.4), results are higher than the previous in terms of precision but lower in case of recall since the captured entities are accurately annotated but lots of False Negatives are present because our system does not capture overlapping entities and in this case one of the entities will not be tagged which substantially decreases recall.

As it can be inferred, most of the errors come in the form of bad boundary detection, which in general causes significantly lower results considering exact matches. With the partial match criteria, the boundary detection importance is decreased, obtaining remarkable results which suggests that with a better boundary detection, the results would be substantially greater.

4.1.4.2 COVID-19 MLIA

Annotation reliability is low because just one annotator carried out the manual annotation process since no consensus has been done on the captured entities and its detected boundaries, making that in some occasions we have found some inconsistencies or entities with doubtful boundaries. Problems related to boundaries have

been downplayed in partial matching criteria, making that recall is directly associated with not captured entities by the system and precision with errors in these captured entities. Comparing the results between exact and partial match criteria, we can see that the results are higher in the later case. Nevertheless, the difference is not as high as in the results obtained in PGxCorpus, which means that not so much of the errors are due to boundary detection problems. Most of the errors come in two ways: On the one hand, a great part of the errors comes as False Positives which it is translated in lower precision. These are mainly produced by entities predicted by the system which does not appear in the ground truth. In some occasions this are due to errors in the system, but in most occasions these are because entities that should have been annotated as a disease are not annotated. For example, "*Acute Respiratory Syndrome*" which is detected by the proposed system is not found in ground truth which makes that FP increases. Another examples are "*Kawasaki disease*", "*Middle East Respiratory Syndrome*", "*immune deficiency*", etc. . . In the other hand, another source of errors comes from signs and symptoms like "*sneeze*", "*cough*", "*high temperature*", etc. . . which since they are not specifically diseases are rarely detected by the proposed system, which decreases recall.

4.2 Normalization Results

Results obtained in the normalization process are difficult to assess since there are multiple sources based on which we perform our entity linking, making that a corpus for testing that should match these requirements in terms of the considered sources. This makes that for the proposed normalization sources, no test corpus will be adapted, making that these results could not be established based on that test corpus. This diversity of sources makes that some unification of concepts should be done in order to make possible for these corpus to evaluate multiple sources. This could be established through the establishment of taxonomy categories for the linked entities and not just an identifier in a certain database and subsequently using this taxonomy for assessing the normalization performance of a given system. This highlights the lack of gold-standard corpus which uses this kind of linking, being present for the most part of the corpus just an identifier to a certain database.

Nevertheless, some observations have been discussed to illustrate how the normalization step works. Worse results were generally observed in the genetic class, maybe because it is a broader semantic class and includes multiple subclasses making that the retrieved terms are not enough to cover all the variability present in that class.

Since in the search of the normalized entity some synonyms are also considered, the results are retrieved even when the entity apparently is not exactly the same allowing some polysemy in our search space. For example, the term "*favilavir*" will be linked to "*Favipiravir*" (CHEBI:134722) since is a synonym. Some terms have not been covered since the possibilities in each of the classes are enormous, overall in the genetic class, making that some entities will not be linked or will be linked either with more general terms or with erroneous terms. In the genetic class, some entities can be referred to multiple organisms, making that the identifier of the concept changes between organisms. This makes that in some occasions the term is erroneously assigned to other organisms concepts since the normalization does not include any way of knowing the organism to which the entity is referred.

Another aspect to take into account is the bad linking in more general entities. For example, the entity "*interferon*" is linked with "*interferon lambda*". This is usually found in occasions where the general term is not present in the database. It was also observed some inconsistencies in some polisemic words. For instance, for the entity "SARS-CoV" we assign "*Severe Acute Respiratory Syndrome*" (CUI:C1175175) but for "SARS" we assign "*Pneumonia due to SARS-associated coronavirus*" (CUI:C1260415) which is partially correct. Since the searched term is slightly different, this causes that the retrieved term is different. Solutions for this kind of inconsistencies could come through a more intensive mapping of concepts in database.

Errors in the previous step, NER, are spread along the normalization producing cascading errors. As it was seen in the review, this is faced in some models like TaggerOne [81] jointly performing both steps at the same time. Nevertheless, the way this system is proposed does not allow to perform it jointly.

4.3 CORD-19 Corpus Annotation Results

The annotation of this corpus (June Edition) is a long process since 6,5M paragraphs need to be annotated. For this purpose, a server in OEG-DIA was used, which is composed by a 32 cores Intel Xeon (Cascade Lake) jointly with 256GB RAM. The lack of a GPU made that process was considerably slower since it had to be carried out by the CPU (although 32 cores are working, the processing is still slow), which is significantly lower for tensor calculations required in transformer-based models like BERT. Paragraphs were processed in a rhythm of approximately 0,4s/paragraph, making that the total approximate time required for the process under the given infrastructure was of 700h which are around 30 days. Normalization has also been done in the processing of each paragraph, but since Solr is a very fast solution for retrieving information and the database is located in localhost, the time required for this subsequent process is insignificant (on the order of milliseconds) compared to the time taken by the previous BioBERT models.

All these made that a subset conformed by around a third of the annotations (2125000) has been considered for discussion of results and statistics¹. The rest of the paragraphs will continue its annotation for further use in Drugs4Covid platform.

In Table 4.6 some statistics are given per each of the entity classes. From the 2125000 considered paragraphs, the entity class for which more paragraphs were annotated with at least one entity was the disease class with 54,96% of paragraphs annotated, far from the rest which were 17,57% for chemicals and 20,11% for genetics. This shows a major mention of diseases which is logical being publications related to COVID-19. All articles speak about the infectious disease or related diseases, but they do not necessarily cover chemicals and genetics related to it. The entity class that on average has more entities per paragraph was genetics with 4,16 entities/paragraph, which significantly more than in diseases and chemicals with 3,15 and 2,73 respectively. This suggests a higher concentration of genetic terms in the paragraphs which speak about genetics. In relation to the normalization process, genetic class was the class in which more terms could not be normalized with a 79,51% of entities normalized. This was significantly better for diseases and chemicals with almost the same percentage of normalized terms, 87% and 87,51% respectively. This reveals

¹All these stats can be viewed in <http://librairy.linkeddata.es/solr/#/cord19-paragraphs/>

Evaluation and Conclusions

which extension of terms in the genetic index for normalization could not be enough to address the large number of possibilities which genetic terms present.

	Paragraphs	% Annotated paragraphs	Total ents	Avg Ents/Paragraph	Total ents normalized	% Normalized ents
Diseases	1167906	54,96	3677866	3,15	3199688	87
Chemicals	373295	17,57	1020795	2,73	893324	87,51
Genetics	427244	20,11	1778047	4,16	1413816	79,51

Table 4.6: Statistics about annotation and normalization of 2125000 processed paragraphs from CORD-19 corpus [135]

In relation to the most widely captured entities, Table 4.7 shows these for each entity class. Jointly with the top words and its position in the top for each entity in the captured entities, the occurrences of these words are given. This allows us to have an idea about how 1:1 are normalized terms regarding candidate terms. This is because if we have, for instance, the word "*tocilizumab*" as candidate and we normalize it with the same word, then a relation 1:1 is present. Nevertheless, the presence of multiple synonyms to refer to this word such as "*Atlizumab*" significantly affect these relations. For that reason, this will only partially reflect the relation and it is just used to get an idea from above. In all classes, in top positions we can find more general words like "*covid*" in diseases, acid in chemicals and protein in genetics. In top positions we can also find general tokens such as numbers like "*19*" or "*2*" which are very present due to the words COVID-19 and SARS-CoV-2 in the case of diseases. As we go down in the top, more specific words begin to appear.

In the disease class, the first words are highly related to COVID terms ("*covid*", "*infection*", "*sars*", "*syndrome*") but as we go down another specific diseases are present like influenza, pneumonia, diabetes, anxiety, etc. . . In relation to the difference between occurrences, we can see how most number of occurrences in annotated and normalized words are very similar. It is worth highlighting a huge difference both in words "*sars*" and "*syndrome*", this is mainly because the terms with the candidate word "*sars*" are usually classified as "*Severe acute respiratory syndrome coronavirus 2*" (NCBI:2697049). The word "*mers*" is normalized as "*Middle East respiratory syndrome-related coronavirus*" and therefore "*mers*" will not be present as a word in normalized terms.

In chemical class, the first positions correspond to general terms in the chemical field like "*acid*", "*amino*" or "*oxygen*". As we move down in the top more specific words begin to appear, as it is the case of specific drugs related with the treatment of COVID like "*hydroxychloroquine*", "*tocilizumab*", "*lopinavir*", etc. . . Occurrences are very similar between candidate and normalized words, which could suggest a high relation between candidates and normalized words.

In the genetic class, the first positions also correspond to very general words like "*protein*", "*receptor*", "*gene*", etc. . . In lower positions specific words appear which are highly related with SARS, this is the case of "*ACE2*", "*IGG*", "*IFN*", "*TNF*", etc. . . In relation to the occurrences most differences are low with the exception of the word "*gene*" which is significantly lower in normalized words. This is because in cases like "*ORF8a gene*", in most occasions in its normalization the word "*gene*" is dismissed resulting in "*ORF8a*".

In the additional normalization of genetic COVID evidences, 103634 paragraphs con-

4.3. CORD-19 Corpus Annotation Results

	Position in Top	Most common words		
		Word	Occurrences in annotated ents	Occurrences in normalized ents
Disease	1	covid	364362	381994
	3	infection	228342	170677
	4	sars	187477	36967
	13	syndrome	47614	243265
	14	influenza	45828	45845
	16	pneumonia	42474	77336
	17	cancer	41533	37915
	19	fever	36073	40917
	20	inflammation	34072	28935
	29	diabetes	24840	26631
	30	mers	24710	-
	31	anxiety	23958	23399
Chemical	1	acid	36567	43293
	3	amino	27359	28947
	4	oxygen	23581	23653
	10	glucose	10144	10282
	19	phosphate	7263	9232
	20	hydroxychloroquine	7202	9543
	21	vitamin	6471	6606
	30	tocilizumab	4433	4433
	34	lopinavir	4238	4503
	36	ritonavir	4130	4388
	37	azithromycin	4095	4092
	38	creatinine	3916	3918
Genetic	1	protein	75149	62529
	5	receptor	28013	32499
	7	gene	26996	4100
	8	ace2	25216	23590
	12	igg	22010	15156
	13	sars	21517	32096
	15	ifn	20040	16754
	20	spike	16098	9362
	22	antibody	15190	-
	27	tnf	13494	12913
	37	angiotensin	9614	11626
	43	igm	8838	7410

Table 4.7: Most common words in each entity class in CORD-19 corpus

tained an evidence (4,88%) with a total of 160910 evidences (1,55 entities/paragraph). The evidences which most times were detected were ACE2 (20531 occurrences) , which is the functional receptor for SARS-CoV-2, Spike protein S2 and S1 (6710 and 2581 occurrences respectively) , which is a structure protein located in

the virus coat, CRP (4553 occurrences), which is a protein present due to inflammatory reactions, TMPRSS2 (4546 occurrences), which is a cell surface protein primarily expressed by endothelial cells across the respiratory tract, etc. . .

4.4 Future Work

Throughout the development of the implementation and with the discussion established in the detected errors, a number of future lines can be established for improvements in further developments. The following are the future works in which it has been observed that the proposed system could improve:

- **Improve boundary detection:** A great part of the errors were observed to be due to an incorrect detection of the boundaries of a detected entity. The proposed post-processing step proposed for this boundary detection is too simple and is just limited to situations where just a part of a word is tagged. Complementary steps could be added in this step for joining multiple words which should be captured jointly. For this purpose, it could be established some linguistic rules which using a POS tagging could associate captured terms improving boundary detection.
- **Allow overlapping:** The framework used (Spacy) does not mainly allow overlappings. Until now, we were considering the larger possible entities which should correspond to the most complete entity and dismissing nested entities since they are usually part of the description of the larger entity. For example, with "*p53 mutagenic cancer*", the proposed system would capture the entire text span as a disease dismissing "*p53*" which is a genetic entity. For some purposes, it could be better to capture the most complete entity but for others also the nested entities found. Depending on that purpose, the system could be extended with some extension attributes for containing nested entities in Spacy.
- **Enhance normalization terms database:** On the one hand, the presence of some different terms for referring same concepts causes some inconsistencies in further normalized entities. Some mappings were established in the processing of these terms, but more intensive mappings will be needed to enhance inconsistency retrievals. On the other hand, the number of retrieved entities for each class is high but nevertheless not enough to cover all possibilities. More sources should be established, overall for covering some semantic subclasses in the genetic class.

Another future works could come not only as a way of improving results but as a way of extending the NLP system. These are as follows:

- **More entity classes:** Until now, this proposal is just implemented for the most widely adopted entities: diseases, chemicals and genetics. Further extensions could come as an implementation of new models for new entity classes like organisms or cell lines, which are classes that have also been widely adopted in literature models.
- **New tasks:** BioBERT model could be easily fine-tuned for carrying out complementary tasks such as relation extraction, which could help to associate the captured entities to supply a more advanced information extraction.

4.5 Conclusions

A walk through the progress that has been made in recent years was studied in the state-of-the-art review. This allowed us to know how NER task has been performed and how it has evolved until the current state-of-the-art, which is based on pretrained BERT models. Based on all this, we were later able to implement a solution based on this state-of-the-art with the use of BioBERT models. These models were fine-tuned in Gold Standard Corpus which allowed them to leverage all the biomedical knowledge underlying them to perform a BioNER task in a given entity class. With this process, we were able to create a system composed by three different models in which each of them is an expertise in a target entity class: diseases, chemicals and genetics. We have also performed a normalization step in which the entities retrieved by the model are linked to identifiers in some curated databases such as MESH, UMLS, CHEBI, etc. . . This normalization has been done through an inverse index search on a set of retrieved terms from multiple sources which allows us to customize with other sources or in the way these terms are represented. The resultant proposed system has been then applied in two practical cases. A first one where it takes part in a web platform where given a text, the system processes it and represents results and a second one where we have used the system to annotate and normalize a CORD-19 corpus. Therefore, the web platform will ease the research community to freely handle these tasks with state-of-the-art models without worrying about the system setup. The annotation of CORD-19 has helped in the information extraction of large amounts of data which will be useful in subsequent tasks in Drugs4Covid platform [28] or in any other required platform.

Results obtained suggested that the system performed well in most occasions except for some boundary detection which has a significant range of improvement. Normalization could also improve overall on the side of the indexed terms. In conclusion, despite the fact that the system has scope for improvement, the results suggest that the system could be successfully applied in a process where information extraction in the biomedical field is needed.

Bibliography

- [1] Anatomical therapeutic chemical classification - summary | ncbo bioportal. <https://bioportal.bioontology.org/ontologies/ATC>. (Accessed on 05/19/2021).
- [2] Chemical entities of biological interest ontology - summary | ncbo bioportal. <https://bioportal.bioontology.org/ontologies/CHEBI>. (Accessed on 05/19/2021).
- [3] Dissecting bert part 1: The encoder | by miguel romero calvo | dissecting bert | medium. <https://medium.com/dissecting-bert/dissecting-bert-part-1-d3c3d495cdb3>. (Accessed on 05/13/2021).
- [4] Gene ontology - summary | ncbo bioportal. <https://bioportal.bioontology.org/ontologies/GO>. (Accessed on 05/19/2021).
- [5] Human disease ontology - summary | ncbo bioportal. <https://bioportal.bioontology.org/ontologies/DOID>. (Accessed on 05/19/2021).
- [6] International classification of diseases, version 10 - clinical modification - summary | ncbo bioportal. <https://bioportal.bioontology.org/ontologies/ICD10CM>. (Accessed on 05/19/2021).
- [7] Medical subject headings - summary | ncbo bioportal. <https://bioportal.bioontology.org/ontologies/MESH>. (Accessed on 05/19/2021).
- [8] Ontology of genes and genomes - summary | ncbo bioportal. <https://bioportal.bioontology.org/ontologies/OGG>. (Accessed on 05/23/2021).
- [9] Protein ontology - summary | ncbo bioportal. <https://bioportal.bioontology.org/ontologies/PR>. (Accessed on 05/19/2021).
- [10] Pubchem. <https://pubchem.ncbi.nlm.nih.gov/>. (Accessed on 05/21/2021).
- [11] Standoff format - brat rapid annotation tool. <https://brat.nlplab.org/standoff.html>. (Accessed on 06/13/2021).
- [12] Survey - bert. https://msank00.github.io/blog/2020/04/13/blog_607_Survey_BERT. (Accessed on 05/13/2021).
- [13] The comparative toxicogenomics database | ctd. <http://ctdbase.org/>, . (Accessed on 05/19/2021).
- [14] The illustrated bert, elmo, and co. (how nlp cracked transfer learning) –

- jay alammar – visualizing machine learning one concept at a time. <https://jalammar.github.io/illustrated-bert/>, . (Accessed on 05/13/2021).
- [15] The illustrated transformer – jay alammar – visualizing machine learning one concept at a time. <https://jalammar.github.io/illustrated-transformer/>, . (Accessed on 05/13/2021).
- [16] Visualizing a neural machine translation model (mechanics of seq2seq models with attention) – jay alammar – visualizing machine learning one concept at a time. <https://jalammar.github.io/visualizing-neural-machine-translation-mechanics-of-seq2seq-models-with-at> (Accessed on 05/13/2021).
- [17] covid19-mlia / organizers-task1 / ground-truth / round1 — bitbucket. <https://bitbucket.org/covid19-mlia/organizers-task1/src/master/ground-truth/round1/>. (Accessed on 06/14/2021).
- [18] django-haystack/pysolr: Pysolr — python solr client. <https://github.com/django-haystack/pysolr>. (Accessed on 05/31/2021).
- [19] explosion/spacy: industrial-strength natural language processing (nlp) in python. <https://github.com/explosion/spaCy>. (Accessed on 05/31/2021).
- [20] huggingface/transformers: transformers: State-of-the-art natural language processing for pytorch, tensorflow, and jax. <https://github.com/huggingface/transformers>. (Accessed on 05/31/2021).
- [21] pallets/flask: The python micro framework for building web applications. <https://github.com/pallets/flask>. (Accessed on 05/31/2021).
- [22] pytorch/pytorch: Tensors and dynamic neural networks in python with strong gpu acceleration. <https://github.com/pytorch/pytorch>. (Accessed on 05/31/2021).
- [23] Saber A Akhondi, Alexander G Klenner, Christian Tyrchan, Anil K Manchala, Kiran Boppana, Daniel Lowe, Marc Zimmermann, Sarma ARP Jagarlapudi, Roger Sayle, Jan A Kors, et al. Annotated chemical patent corpus: a gold standard for text mining. *PloS one*, 9(9):e107477, 2014.
- [24] Emily Alsentzer, John R Murphy, Willie Boag, Wei-Hung Weng, Di Jin, Tristan Naumann, and Matthew McDermott. Publicly available clinical bert embeddings. *arXiv preprint arXiv:1904.03323*, 2019.
- [25] Michael Ashburner, Catherine A Ball, Judith A Blake, David Botstein, Heather Butler, J Michael Cherry, Allan P Davis, Kara Dolinski, Selina S Dwight, Janan T Eppig, et al. Gene ontology: tool for the unification of biology. *Nature genetics*, 25(1):25–29, 2000.
- [26] Muhammad Aslam, Jae-Myeong Lee, Hyung-Seung Kim, Seung-Jae Lee, and Sugwon Hong. Deep learning models for long-term solar radiation forecasting considering microgrid installation: A comparative study. *Energies*, 13(1):147, 2020.
- [27] Michael Bada, Miriam Eckert, Donald Evans, Kristin Garcia, Krista Shipley, Dmitry Sitnikov, William A Baumgartner, K Bretonnel Cohen, Karin Verspoor,

- Judith A Blake, et al. Concept annotation in the craft corpus. *BMC bioinformatics*, 13(1):1–20, 2012.
- [28] Carlos Badenes-Olmedo, David Chaves-Fraga, María Poveda-Villalón, Ana Iglesias-Molina, Pablo Calleja, Socorro Bernardos, Patricia Martín-Chozas, Alba Fernández-Izquierdo, Elvira Amador-Domínguez, Paola Espinoza-Arias, et al. Drugs4covid: Drug-driven knowledge exploitation based on scientific publications. *arXiv preprint arXiv:2012.01953*, 2020.
- [29] Shweta Bagewadi, Tamara Bobić, Martin Hofmann-Apitius, Juliane Fluck, and Roman Klinger. Detecting mirna mentions and relations in biomedical literature. *F1000Research*, 3, 2014.
- [30] Andrew L Beam, Benjamin Kompa, Allen Schmaltz, Inbar Fried, Griffin Weber, Nathan Palmer, Xu Shi, Tianxi Cai, and Isaac S Kohane. Clinical concept embeddings learned from massive sources of multimodal medical data. In *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, volume 25, pages 295–306. World Scientific, 2020.
- [31] Iz Beltagy, Kyle Lo, and Arman Cohan. Scibert: A pretrained language model for scientific text. *arXiv preprint arXiv:1903.10676*, 2019.
- [32] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. A neural probabilistic language model. *The journal of machine learning research*, 3:1137–1155, 2003.
- [33] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017.
- [34] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.
- [35] David Campos, Sérgio Matos, and José Luís Oliveira. Biomedical named entity recognition: a survey of machine-learning tools. *Theory and Applications for Advanced Text Mining*, 11:175–195, 2012.
- [36] David Campos, Sérgio Matos, and José Luís Oliveira. Gimli: open source and high-performance biomedical name recognition. *BMC bioinformatics*, 14(1):1–14, 2013.
- [37] Qingyu Chen, Yifan Peng, and Zhiyong Lu. Biosentvec: creating sentence embeddings for biomedical texts. In *2019 IEEE International Conference on Healthcare Informatics (ICHI)*, pages 1–5. IEEE, 2019.
- [38] Billy Chiu, Gamal Crichton, Anna Korhonen, and Sampo Pyysalo. How to train good word embeddings for biomedical nlp. In *Proceedings of the 15th workshop on biomedical natural language processing*, pages 166–174, 2016.
- [39] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

-
- [40] Youngduck Choi, Chill Yi-I Chiu, and David Sontag. Learning low-dimensional representations of medical concepts. *AMIA Summits on Translational Science Proceedings*, 2016:41, 2016.
 - [41] Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167, 2008.
 - [42] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *Journal of machine learning research*, 12(ARTICLE):2493–2537, 2011.
 - [43] Gamal Crichton, Sampo Pyysalo, Billy Chiu, and Anna Korhonen. A neural network multi-task learning approach to biomedical named entity recognition. *BMC bioinformatics*, 18(1):1–14, 2017.
 - [44] Thanh Hai Dang, Hoang-Quynh Le, Trang M Nguyen, and Sinh T Vu. D3ner: biomedical named entity recognition using crf-bilstm improved with fine-tuned embeddings of various linguistic information. *Bioinformatics*, 34(20):3539–3546, 2018.
 - [45] Lance De Vine, Guido Zuccon, Bevan Koopman, Laurianne Sitbon, and Peter Bruza. Medical semantic similarity with a neural language model. In *Proceedings of the 23rd ACM international conference on conference on information and knowledge management*, pages 1819–1822, 2014.
 - [46] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.
 - [47] Rezarta Islamaj Doğan, Robert Leaman, and Zhiyong Lu. Ncbi disease corpus: a resource for disease name recognition and concept normalization. *Journal of biomedical informatics*, 47:1–10, 2014.
 - [48] John R Firth. A synopsis of linguistic theory, 1930-1955. *Studies in linguistic analysis*, 1957.
 - [49] Zachary N Flamholz, Lyle H Ungar, and Gary Eric Weissman. Word embeddings trained on published case reports are lightweight, effective for clinical tasks, and free of protected health information. *medRxiv*, page 19013268, 2019.
 - [50] Sylvain Gaudan, Harald Kirsch, and Dietrich Rebholz-Schuhmann. Resolving abbreviations to their senses in medline. *Bioinformatics*, 21(18):3658–3664, 2005.
 - [51] Sebastian Gehrmann, Franck Dernoncourt, Yeran Li, Eric T Carlson, Joy T Wu, Jonathan Welt, John Foote Jr, Edward T Moseley, David W Grant, Patrick D Tyler, et al. Comparing deep learning and concept extraction based methods for patient phenotyping from clinical narratives. *PloS one*, 13(2):e0192360, 2018.
 - [52] Martin Gerner, Goran Nenadic, and Casey M Bergman. Linnaeus: a species name identification system for biomedical literature. *BMC bioinformatics*, 11(1):1–17, 2010.
 - [53] John M Giorgi and Gary D Bader. Transfer learning for biomedical named entity recognition with neural networks. *Bioinformatics*, 34(23):4087–4094, 2018.

- [54] John M Giorgi and Gary D Bader. Towards reliable named entity recognition in the biomedical domain. *Bioinformatics*, 36(1):280–286, 2020.
- [55] Tatyana Goldberg, Shrikant Vinchurkar, Juan Miguel Cejuela, Lars Juhl Jensen, and Burkhard Rost. Linked annotations: a middle ground for manual curation of biomedical databases and text corpora. In *BMC proceedings*, volume 9, pages 1–3. BioMed Central, 2015.
- [56] Rodrigo Rafael Villarreal Goulart, Vera Lúcia Strube de Lima, and Clarissa Castellã Xavier. A systematic review of named entity recognition in biomedical texts. *Journal of the Brazilian Computer Society*, 17(2):103–116, 2011.
- [57] Archana Goyal, Vishal Gupta, and Manish Kumar. Recent named entity recognition and classification techniques: a systematic review. *Computer Science Review*, 29:21–43, 2018.
- [58] Alex Graves and Jürgen Schmidhuber. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural networks*, 18(5-6):602–610, 2005.
- [59] Mourad Gridach. Character-level neural network for biomedical named entity recognition. *Journal of biomedical informatics*, 70:85–91, 2017.
- [60] Ralph Grishman and Beth Sundheim. Message understanding conference- 6: A brief history. In *COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics*, 1996. URL <https://www.aclweb.org/anthology/C96-1079>.
- [61] Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. Domain-specific language model pretraining for biomedical natural language processing. *arXiv preprint arXiv:2007.15779*, 2020.
- [62] Harsha Gurulingappa, Roman Klinger, Martin Hofmann-Apitius, and Juliane Fluck. An empirical evaluation of resources for the identification of diseases and adverse effects in biomedical literature. In *2nd Workshop on Building and evaluating resources for biomedical text mining (7th edition of the Language Resources and Evaluation Conference)*, 2010.
- [63] Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A Smith. Don’t stop pretraining: Adapt language models to domains and tasks. *arXiv preprint arXiv:2004.10964*, 2020.
- [64] Maryam Habibi, Leon Weber, Mariana Neves, David Luis Wiegandt, and Ulf Leser. Deep learning with word embeddings improves biomedical named entity recognition. *Bioinformatics*, 33(14):i37–i48, 2017.
- [65] Zellig S Harris. Distributional structure. *Word*, 10(2-3):146–162, 1954.
- [66] Yongqun He, Yue Liu, and Bin Zhao. Ogg: a biological ontology for representing genes and genomes in specific organisms. In *ICBO*, pages 13–20. Citeseer, 2014.
- [67] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

-
- [68] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*, 2016.
- [69] Meizhi Ju, Makoto Miwa, and Sophia Ananiadou. A neural layered model for nested named entity recognition. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1446–1459, 2018.
- [70] Suwisa Kaewphan, Sofie Van Landeghem, Tomoko Ohta, Yves Van de Peer, Filip Ginter, and Sampo Pyysalo. Cell line name recognition in support of the identification of synthetic lethality in cancer from text. *Bioinformatics*, 32(2): 276–282, 2016.
- [71] Donghyeon Kim, Jinhyuk Lee, Chan Ho So, Hwisang Jeon, Minbyul Jeong, Yonghwa Choi, Wonjin Yoon, Mujeen Sung, and Jaewoo Kang. A neural named entity recognition and multi-type normalization tool for biomedical text mining. *IEEE Access*, 7:73729–73740, 2019.
- [72] Jin-Dong Kim, Tomoko Ohta, Yoshimasa Tsuruoka, Yuka Tateisi, and Nigel Collier. Introduction to the bio-entity recognition task at jnlpba. In *Proceedings of the international joint workshop on natural language processing in biomedicine and its applications*, pages 70–75. Citeseer, 2004.
- [73] Aris Kosmopoulos, Ion Androutsopoulos, and Georgios Paliouras. Biomedical semantic indexing using dense word vectors in bioasq. *J BioMed Semant Suppl BioMedl Inf Retr*, 3410:959136040–1510456246, 2015.
- [74] Martin Krallinger, Obdulia Rabal, Florian Leitner, Miguel Vazquez, David Salgado, Zhiyong Lu, Robert Leaman, Yanan Lu, Donghong Ji, Daniel M Lowe, et al. The chemdner corpus of chemicals and drugs and its annotation principles. *Journal of cheminformatics*, 7(1):1–17, 2015.
- [75] Martin Krallinger, Obdulia Rabal, Analia Lourenço, Martin Perez Perez, Gael Perez Rodriguez, Miguel Vazquez, Florian Leitner, Julen Oyarzabal, and Alfonso Valencia. Overview of the chemdner patents task. In *Proceedings of the fifth BioCreative challenge evaluation workshop*, pages 63–75, 2015.
- [76] Taku Kudo and John Richardson. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *arXiv preprint arXiv:1808.06226*, 2018.
- [77] Seth Kulick, Ann Bies, Mark Liberman, Mark Mandel, Ryan McDonald, Martha Palmer, Andrew Schein, Lyle Ungar, Scott Winters, and Peter White. Integrated annotation for biomedical information extraction. In *HLT-NAACL 2004 Workshop: Linking Biological Literature, Ontologies and Databases*, pages 61–68, 2004.
- [78] John Lafferty, Andrew McCallum, and Fernando CN Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. 2001.
- [79] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360*, 2016.

- [80] Robert Leaman and Graciela Gonzalez. Banner: an executable survey of advances in biomedical named entity recognition. In *Biocomputing 2008*, pages 652–663. World Scientific, 2008.
- [81] Robert Leaman and Zhiyong Lu. Taggerone: joint named entity recognition and normalization with semi-markov models. *Bioinformatics*, 32(18):2839–2846, 2016.
- [82] Robert Leaman, Christopher Miller, and Graciela Gonzalez. Enabling recognition of diseases in biomedical text with machine learning: corpus and benchmark. In *Proceedings of the 2009 Symposium on Languages in Biology and Medicine*, volume 82, 2009.
- [83] Robert Leaman, Chih-Hsuan Wei, and Zhiyong Lu. tmchem: a high performance approach for chemical named entity recognition and normalization. *Journal of cheminformatics*, 7(1):1–10, 2015.
- [84] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240, 2020.
- [85] Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. End-to-end neural coreference resolution. *arXiv preprint arXiv:1707.07045*, 2017.
- [86] Joël Legrand, Romain Gogdemir, Cédric Bousquet, Kevin Dalleau, Marie-Dominique Devignes, William Digan, Chia-Ju Lee, Ndeye-Coumba Ndiaye, Nadine Petitpain, Patrice Ringot, et al. Pgxcopus, a manually annotated corpus for pharmacogenomics. *Scientific data*, 7(1):1–13, 2020.
- [87] Vladimir I Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710. Soviet Union, 1966.
- [88] Jiao Li, Yueping Sun, Robin J Johnson, Daniela Sciaky, Chih-Hsuan Wei, Robert Leaman, Allan Peter Davis, Carolyn J Mattingly, Thomas C Wieggers, and Zhiyong Lu. Biocreative v cdr task corpus: a resource for chemical disease relation extraction. *Database*, 2016, 2016.
- [89] Jing Li, Aixun Sun, Jianglei Han, and Chenliang Li. A survey on deep learning for named entity recognition. *IEEE Transactions on Knowledge and Data Engineering*, 2020.
- [90] Lishuang Li, Liuke Jin, Zhenchao Jiang, Dingxin Song, and Degen Huang. Biomedical named entity recognition based on extended recurrent neural networks. In *2015 IEEE International Conference on bioinformatics and biomedicine (BIBM)*, pages 649–652. IEEE, 2015.
- [91] Edward Loper and Steven Bird. Nltk: The natural language toolkit. *arXiv preprint cs/0205028*, 2002.
- [92] Ling Luo, Zhihao Yang, Pei Yang, Yin Zhang, Lei Wang, Hongfei Lin, and Jian Wang. An attention-based bilstm-crf approach to document-level chemical named entity recognition. *Bioinformatics*, 34(8):1381–1388, 2018.

-
- [93] Chen Lyu, Bo Chen, Yafeng Ren, and Donghong Ji. Long short-term memory rnn for biomedical named entity recognition. *BMC bioinformatics*, 18(1):1–11, 2017.
 - [94] Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, pages 55–60, 2014.
 - [95] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.
 - [96] Tomáš Mikolov, Wen-tau Yih, and Geoffrey Zweig. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 conference of the north american chapter of the association for computational linguistics: Human language technologies*, pages 746–751, 2013.
 - [97] Maria Mitrofan and Radu Ion. Adapting the ttl romanian pos tagger to the biomedical domain. In *BiomedicalNLP@ RANLP*, pages 8–14, 2017.
 - [98] SPFGH Moen and Tapio Salakoski² Sophia Ananiadou. Distributional semantics resources for biomedical text processing. *Proceedings of LBM*, pages 39–44, 2013.
 - [99] Behrang Mohit. Named entity recognition. In *Natural language processing of semitic languages*, pages 221–245. Springer, 2014.
 - [100] David Nadeau and Satoshi Sekine. A survey of named entity recognition and classification. *Linguisticae Investigationes*, 30(1):3–26, 2007.
 - [101] Darren A Natale, Cecilia N Arighi, Judith A Blake, Jonathan Bona, Chuming Chen, Sheng-Chih Chen, Karen R Christie, Julie Cowart, Peter D’Eustachio, Alexander D Diehl, et al. Protein ontology (pro): enhancing and scaling up the representation of protein entities. *Nucleic acids research*, 45(D1):D339–D346, 2017.
 - [102] Mark Neumann, Daniel King, Iz Beltagy, and Waleed Ammar. ScispaCy: Fast and Robust Models for Biomedical Natural Language Processing. In *Proceedings of the 18th BioNLP Workshop and Shared Task*, pages 319–327, Florence, Italy, August 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-5034. URL <https://www.aclweb.org/anthology/W19-5034>.
 - [103] Mariana Neves, Alexander Damaschun, Andreas Kurtz, and Ulf Leser. Annotating and evaluating text for stem cell research. In *Proceedings of the Third Workshop on Building and Evaluation Resources for Biomedical Text Mining (BioTxtM 2012) at Language Resources and Evaluation (LREC). Istanbul, Turkey*, pages 16–23. Citeseer, 2012.
 - [104] Tomoko Ohta, Sampo Pyysalo, Rafal Rak, Andrew Rowley, Hong-Woo Chun, Sung-Jae Jung, Sung-Pil Choi, Sophia Ananiadou, and Jun’ichi Tsujii. Overview of the pathway curation (pc) task of bionlp shared task 2013. In *Proceedings of the BioNLP Shared Task 2013 Workshop*, pages 67–75, 2013.
 - [105] Evangelos Pafilis, Sune P Frankild, Lucia Fanini, Sarah Faulwetter, Christina Pavloudi, Aikaterini Vasileiadou, Christos Arvanitidis, and Lars Juhl Jensen.

- The species and organisms resources for fast and accurate identification of taxonomic names in text. *PloS one*, 8(6):e65390, 2013.
- [106] Yifan Peng, Shankai Yan, and Zhiyong Lu. Transfer learning in biomedical natural language processing: an evaluation of bert and elmo on ten benchmarking datasets. *arXiv preprint arXiv:1906.05474*, 2019.
- [107] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.
- [108] Nadeesha Perera, Matthias Dehmer, and Frank Emmert-Streib. Named entity recognition and relation detection for biomedical information extraction. *Frontiers in Cell and Developmental Biology*, 8:673, 2020.
- [109] Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*, 2018.
- [110] Sampo Pyysalo and Sophia Ananiadou. Anatomical entity mention recognition at literature scale. *Bioinformatics*, 30(6):868–875, 2014.
- [111] Sampo Pyysalo, Filip Ginter, Juho Heimonen, Jari Björne, Jorma Boberg, Jouni Järvinen, and Tapio Salakoski. Bioinfer: a corpus for information extraction in the biomedical domain. *BMC bioinformatics*, 8(1):1–24, 2007.
- [112] Sampo Pyysalo, Tomoko Ohta, Rafal Rak, Dan Sullivan, Chunhong Mao, Chunxia Wang, Bruno Sobral, Jun’ichi Tsujii, and Sophia Ananiadou. Overview of the id, epi and rel tasks of bionlp shared task 2011. In *BMC bioinformatics*, volume 13, pages 1–26. Springer, 2012.
- [113] Sampo Pyysalo, Tomoko Ohta, Rafal Rak, Andrew Rowley, Hong-Woo Chun, Sung-Jae Jung, Sung-Pil Choi, Jun’ichi Tsujii, and Sophia Ananiadou. Overview of the cancer genetics and pathway curation tasks of bionlp shared task 2013. *BMC bioinformatics*, 16(10):1–19, 2015.
- [114] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2018.
- [115] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.
- [116] Tim Rocktäschel, Michael Weidlich, and Ulf Leser. Chemspot: a hybrid system for chemical named entity recognition. *Bioinformatics*, 28(12):1633–1640, 2012.
- [117] Lynn Marie Schriml, Cesar Arze, Suvarna Nadendla, Yu-Wei Wayne Chang, Mark Mazaitis, Victor Felix, Gang Feng, and Warren Alden Kibbe. Disease ontology: a backbone for disease semantic integration. *Nucleic acids research*, 40(D1):D940–D946, 2012.
- [118] Mike Schuster and Kaisuke Nakajima. Japanese and korean voice search. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5149–5152. IEEE, 2012.

-
- [119] Mike Schuster and Kuldip K Paliwal. Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing*, 45(11):2673–2681, 1997.
- [120] Ariel S Schwartz and Marti A Hearst. A simple algorithm for identifying abbreviation definitions in biomedical text. In *Biocomputing 2003*, pages 451–462. World Scientific, 2002.
- [121] Isabel Segura Bedmar, Paloma Martínez, and María Herrero Zazo. Semeval-2013 task 9: Extraction of drug-drug interactions from biomedical texts (ddiextraction 2013). Association for Computational Linguistics, 2013.
- [122] Rahul Sharnagat. Named entity recognition: A literature survey. page 27, 2014.
- [123] Larry Smith, Lorraine K Tanabe, Rie Johnson nee Ando, Cheng-Ju Kuo, I-Fang Chung, Chun-Nan Hsu, Yu-Shi Lin, Roman Klinger, Christoph M Friedrich, Kuzman Ganchev, et al. Overview of biocreative ii gene mention recognition. *Genome biology*, 9(2):1–19, 2008.
- [124] Sunghwan Sohn, Donald C Comeau, Won Kim, and W John Wilbur. Abbreviation definition identification based on automatic precision estimates. *BMC bioinformatics*, 9(1):1–10, 2008.
- [125] Pontus Stenetorp, Goran Topić, Sampo Pyysalo, Tomoko Ohta, Jin-Dong Kim, and Jun’ichi Tsujii. Bionlp shared task 2011: Supporting resources. In *Proceedings of BioNLP Shared Task 2011 Workshop*, pages 112–120, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/W11-1816>.
- [126] Mark Stevenson, Yikun Guo, Abdulaziz Alamri, and Robert Gaizauskas. Disambiguation of biomedical abbreviations. In *Proceedings of the BioNLP 2009 Workshop*, pages 71–79, 2009.
- [127] Lorraine Tanabe, Natalie Xie, Lynne H Thom, Wayne Matten, and W John Wilbur. Genetag: a tagged corpus for gene/protein named entity recognition. *BMC bioinformatics*, 6(1):1–7, 2005.
- [128] Hai Long Trieu, Nhung TH Nguyen, Makoto Miwa, and Sophia Ananiadou. Investigating domain-specific information for neural coreference resolution on biomedical texts. In *Proceedings of the BioNLP 2018 workshop*, pages 183–188, 2018.
- [129] Richard Tzong-Han Tsai, Shih-Hung Wu, Wen-Chi Chou, Yu-Chun Lin, Ding He, Jieh Hsiang, Ting-Yi Sung, and Wen-Lian Hsu. Various criteria in the evaluation of biomedical named entity recognition. *BMC bioinformatics*, 7(1): 1–8, 2006.
- [130] Inigo Jauregi Unanue, Ehsan Zare Borzeshi, and Massimo Piccardi. Recurrent neural networks with specialized word embeddings for health-domain named-entity recognition. *Journal of biomedical informatics*, 76:102–109, 2017.
- [131] Özlem Uzuner, Brett R South, Shuying Shen, and Scott L DuVall. 2010 i2b2/va challenge on concepts, assertions, and relations in clinical text. *Journal of the American Medical Informatics Association*, 18(5):552–556, 2011.

- [132] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.
- [133] Karin Verspoor, Antonio Jimeno Yepes, Lawrence Cavedon, Tara McIntosh, Asha Hertzen-Crabb, Zoë Thomas, and John-Paul Plazzer. Annotating the biomedical literature for the human variome. *Database*, 2013, 2013.
- [134] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*, 2018.
- [135] Lucy Lu Wang, Kyle Lo, Yoganand Chandrasekhar, Russell Reas, Jiangjiang Yang, Doug Burdick, Darrin Eide, Kathryn Funk, Yannis Katsis, Rodney Michael Kinney, Yunyao Li, Ziyang Liu, William Merrill, Paul Mooney, Dewey A. Murdick, Devvret Rishi, Jerry Sheehan, Zhihong Shen, Brandon Stilson, Alex D. Wade, Kuansan Wang, Nancy Xin Ru Wang, Christopher Wilhelm, Boya Xie, Douglas M. Raymond, Daniel S. Weld, Oren Etzioni, and Sebastian Kohlmeier. CORD-19: The COVID-19 open research dataset. In *Proceedings of the 1st Workshop on NLP for COVID-19 at ACL 2020*, Online, July 2020. Association for Computational Linguistics. URL <https://www.aclweb.org/anthology/2020.nlpccovid19-acl.1>.
- [136] Xu Wang, Chen Yang, and Renchu Guan. A comparative study for biomedical named entity recognition. *International Journal of Machine Learning and Cybernetics*, 9(3):373–382, 2015.
- [137] Xuan Wang, Yu Zhang, Xiang Ren, Yuhao Zhang, Marinka Zitnik, Jingbo Shang, Curtis Langlotz, and Jiawei Han. Cross-type biomedical named entity recognition with deep multi-task learning. *Bioinformatics*, 35(10):1745–1752, 2019.
- [138] Chih-Hsuan Wei, Bethany R Harris, Hung-Yu Kao, and Zhiyong Lu. tmvar: a text mining approach for extracting sequence variants in biomedical literature. *Bioinformatics*, 29(11):1433–1439, 2013.
- [139] Chih-Hsuan Wei, Lon Phan, Juliana Feltz, Rama Maiti, Tim Hefferon, and Zhiyong Lu. tmvar 2.0: integrating genomic variant information from literature with dbsnp and clinvar for precision medicine. *Bioinformatics*, 34(1):80–87, 2018.
- [140] Qikang Wei, Tao Chen, Ruifeng Xu, Yulan He, and Lin Gui. Disease named entity recognition by combining conditional random fields and bidirectional recurrent neural networks. *Database*, 2016, 2016.
- [141] William E Winkler. String comparator metrics and enhanced decision rules in the fellegi-sunter model of record linkage. 1990.
- [142] Kai Xu, Zhenguo Yang, Peipei Kang, Qi Wang, and Wenying Liu. Document-level attention-based bilstm-crf incorporating disease dictionary for disease named entity recognition. *Computers in biology and medicine*, 108:122–132, 2019.
- [143] Vikas Yadav and Steven Bethard. A survey on recent advances in named entity recognition from deep learning models. *arXiv preprint arXiv:1910.11470*, 2019.

- [144] Wonjin Yoon, Chan Ho So, Jinhyuk Lee, and Jaewoo Kang. Collabonet: collaboration of deep neural networks for biomedical named entity recognition. *BMC bioinformatics*, 20(10):55–65, 2019.
- [145] Hong Yu, George Hripcsak, and Carol Friedman. Mapping abbreviations to full forms in biomedical articles. *Journal of the American Medical Informatics Association*, 9(3):262–272, 2002.
- [146] Shaodian Zhang and Noémie Elhadad. Unsupervised biomedical named entity recognition: Experiments with clinical and biological texts. *Journal of biomedical informatics*, 46(6):1088–1098, 2013.
- [147] Yijia Zhang, Qingyu Chen, Zhihao Yang, Hongfei Lin, and Zhiyong Lu. Biowordvec, improving biomedical word embeddings with subword information and mesh. *Scientific data*, 6(1):1–9, 2019.
- [148] Sendong Zhao, Ting Liu, Sicheng Zhao, and Fei Wang. A neural multi-task learning framework to jointly model medical named entity recognition and normalization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 817–824, 2019.
- [149] G. Zhou, J. Zhang, J. Su, D. Shen, and C. Tan. Recognizing names in biomedical texts: a machine learning approach. *Bioinformatics*, 20(7):1178–1190, May 2004. ISSN 1367-4803, 1460-2059. doi: 10.1093/bioinformatics/bth060.
- [150] Qile Zhu, Xiaolin Li, Ana Conesa, and Cécile Pereira. Gram-cnn: a deep learning approach with local context for named entity recognition in biomedical text. *Bioinformatics*, 34(9):1547–1554, 2018.

Appendix

.1 SOTA retrieved publications

Year	Title	Author	DOI
2012	ChemSpot: a hybrid system for chemical named entity recognition [116]	Rocktäschel et al.	https://doi.org/10.1093/bioinformatics/bts183
2013	The SPECIES and ORGANISMS resources for fast and accurate identification of taxonomic names in text [105]	Pafilis et al.	https://doi.org/10.1371/journal.pone.0065390
2013	Unsupervised biomedical named entity recognition: Experiments with clinical and biological texts [146]	Zhang et al.	https://doi.org/10.1016/j.jbi.2013.08.004
2013	Gimli: open source and high-performance biomedical name recognition [36]	Campos et al.	https://doi.org/10.1186/1471-2105-14-54
2013	tmVar: A text mining approach for extracting sequence variants in biomedical literature [138]	Wei et al.	https://doi.org/10.1093/bioinformatics/btt156
2015	tmChem: a high performance approach for chemical named entity recognition and normalization [83]	Leaman et al.	https://doi.org/10.1186/1758-2946-7-S1-S3
2016	Disease named entity recognition by combining conditional random fields and bidirectional recurrent neural networks [140]	Wei et al.	https://doi.org/10.1093/database/baw140
2016	TaggerOne: joint named entity recognition and normalization with semi-Markov Models [81]	Leaman et al.	https://doi.org/10.1093/bioinformatics/btw343

Table 8 continued from previous page

2017	Deep learning with word embeddings improves biomedical named entity recognition [64]	Habibi et al.	https://doi.org/10.1093/bioinformatics/btx228
2017	Character-level neural network for biomedical named entity recognition [59]	Gridach et al.	https://doi.org/10.1016/j.jbi.2017.05.002
2017	A neural network multi-task learning approach to biomedical named entity recognition [43]	Crichton et al.	https://doi.org/10.1186/s12859-017-1776-8
2017	Long short-term memory RNN for biomedical named entity recognition [93]	Lyu et al.	https://doi.org/10.1186/s12859-017-1868-5
2017	Recurrent neural networks with specialized word embeddings for health-domain named-entity recognition [130]	Unanue et al.	https://doi.org/10.1016/j.jbi.2017.11.007
2018	An attention-based BiLSTM-CRF approach to document-level chemical named entity recognition [92]	Luo et al.	https://doi.org/10.1093/bioinformatics/btx761
2018	A Neural Layered Model for Nested Named Entity Recognition [69]	Ju et al.	http://dx.doi.org/10.18653/v1/N18-1131
2018	GRAM-CNN: a deep learning approach with local context for named entity recognition in biomedical text [150]	Zhu et al.	https://doi.org/10.1093/bioinformatics/btx815
2018	Transfer learning for biomedical named entity recognition with neural networks [53]	Giorgi and Bader et al.	https://doi.org/10.1093/bioinformatics/bty449
2018	D3NER: biomedical named entity recognition using CRF-biLSTM improved with fine-tuned embeddings of various linguistic information [44]	Dang et al.	https://doi.org/10.1093/bioinformatics/bty356
2019	Cross-type Biomedical Named Entity Recognition with Deep Multi-Task Learning [137]	Wang et al.	https://doi.org/10.1093/bioinformatics/bty869

Table 8 continued from previous page

2019	A Neural Multi-Task Learning Framework to Jointly Model Medical Named Entity Recognition and Normalization [148]	Zhao et al.	https://doi.org/10.1609/aaai.v33i01.3301817
2020	Towards reliable named entity recognition in the biomedical domain [54]	Giorgi and Bader et al.	https://doi.org/10.1093/bioinformatics/btz504
2019	BioBERT: a pre-trained biomedical language representation model for biomedical text mining [84]	Lee et al.	https://doi.org/10.1093/bioinformatics/btz682
2020	Domain-Specific Language Model Pretraining for Biomedical Natural Language Processing [61]	Gu et al.	https://doi.org/10.1145/3458754
2019	Publicly available clinical BERT embeddings [24]	Alsentzer et al.	http://dx.doi.org/10.18653/v1/W19-1909
2019	SciBERT: A pretrained language model for scientific text [31]	Beltagy et al.	http://dx.doi.org/10.18653/v1/D19-1371
2019	Transfer Learning in Biomedical Natural Language Processing: An Evaluation of BERT and ELMo on Ten Benchmarking Datasets [106]	Peng et al.	http://dx.doi.org/10.18653/v1/W19-5006

Table 8: Retrieved papers in SOTA review