UNIVERSIDAD NACIONAL DE EDUCACIÓN A DISTANCIA (UNED)
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA

DETECTING MOST IMPORTANT SENTENCES IN TRAINING CORPUS FOR NER TASK

TRABAJO DE FIN DE MÁSTER PRESENTADO POR
LUIS ESTEBAN ANDALUZ

DIRIGIDO POR
RAQUEL MARTÍNEZ UNANUE

MÁSTER UNIVERSITARIO EN I.A. AVANZADA: FUNDAMENTOS, MÉTODOS Y APLICACIONES
CURSO 2021-2022
CONVOCATORIA DE SEPTIEMBRE

Agradecimientos

A mi tutora, Raquel, por su indispensable labor repasando esta memoria para conseguir que refleje todo el trabajo realizado y que llegue en tiempo y forma a coger mi último tren para acabar este TFM.

A Javier Gamazo por su inestimable ayuda cuando el caos de la falta de aprendizaje se colaba en mis desarrollos y por ser siempre una fuente de buen criterio e inspiración académica.

A Sofi por aguantar mis noches sin dormir, mis agobios, la ansiedad y mi inquietud sin límite, que sepa devolverte cada minuto de tu apoyo.

Abstract

Name Entity Recognition (NER) consists in the location of a word expression that references to an entity in a text. For the last 25 years, this task have been subject of research given its application in a variety of Natural Language Processing (NLP) tasks. Also, NER for biomedical domain has special interest, as well as difficulties given the heterogeneity and polysemy in some entities such as genes, symptoms and diseases. Although NER systems have improve substantially in the past years thanks to Deep Learning fast development, there is still improvement possibilities and for this reason there are still evaluation campaigns to push the state of art further.

In this work, we have proposed an end-to-end system able to accomplish NER task based on deep transformer, BERT. This system uses BIO-labels, so pre and post processing steps have been also designed and developed from scratch. We will use the eHealth Knowledge Discovery Challenge at IberLEF 2021 as a framework for our development. Using this system, the impact of sentence selection in system training is studied. First, we describe the sentences in corpus given certain morpho-syntactical and semantic extracted features. Later, we train the system with different number of sentences, which have been selected given certain feature criteria, and compare the results with the same number of sentences that have been selected in a random selection. Results show that training with certain sentence can perform better that random selections when small amounts of training data are available.

Finally, we calculate and compare the results of our system on eHealth KD 2021 task, as well as, techniques used in of state-of-art results.

Keywords Name Entity Recognition (NER), biomededical, BIO tags, Deep Transformers, BERT, feature extraction, data-centric AI.

Resumen

El Reconocimiento de Entidades Nominales (NER) consiste en la localización de una expresión textual que hace referencia a una entidad en el texto. Durante los últimos 25 años este problema ha sido sujeto de investigación dada su aplicación en variedad de sistemas de Procesamiento del Lenguaje Natural (NLP). Además, el NER en el dominio biomédico tiene un interés especial, así como dificulades debido a la heterogeneidad y polisemia en algunas entidades como son: genes, síntomas y enfermedades. Aunque los sistemas NER han mejorado sustancialmente en los últimos años gracias al rápido desarrollo del Deep Learning, todavía queda márgen de mejora y, por este motivo, todavía se organizan campañas de evaluación para hacer avanzar el estado del arte.

En este trabajo, hemos propuesto un sistema completo capaz de llevar a cabo la tarea NER basado en el *deep transformer*, BERT. Este sistema utiliza etiquetas BIO, por lo que las etapas de pre y post procesamiento también se han diseñado y desarrollado de cero. Utilizaremos la campaña de evaluación eHealth Knowledge Discovery Challenge at IberLEF 2021 como marco para nuestro desarrollo. Utilizando este sistema, vamos a estudiar el impacto de la selección de oraciones en el entrenamiento del propio sistema. Primero, describimos las oraciones del corpus dados ciertos rasgos morfosintácticos y semánticos. Después, entrenamos el sistema con diferente número de oraciones, que han sido seleccionadas según ciertos criterios de los rasgos, y comparamos los resultados con el mismo número de oraciones que han sido escogidas de forma aleatoria. Los resultados muestran que el entrenamiento con oraciones concretas puede desempeñar mejor que la selección aleatoria cuando poca cantidad de datos de entrenamiento están disponibles.

Finalmente, calculamos y comparamos los resultados de este sistema en la tarea eHealth KD 2021, además de las técnicas utilizadas por los resultados al nivel del estado del arte.

Palabras clave Reconocimiento de entidades nominales (NER), biomédico, etiquetas BIO, Deep Transformers, BERT, extracción de rasgos característicos, AI centrada en datos.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Problem description and motivation

Name Entity Recognition (NER) task was first proposed at *6th Message Understanding Conferences* (MUC-6) conference in 1995 [37]. It was defined as the location of a word of expression that references to an entity in the text. At that time, MUC was focused on structure data from unstructured data, i. e. text. Its main topic was companies, defense and its related activities. NER was appointed as an important part of this more complex task and it was presented as a specific competition. MUC-6 task had three types of entity to be recognized.

- **ENAMEX**: Persons, organizations and locations.
- **TIMEX**: dates and times.
- **NUMEX**: Money, percentages and quantities.

Name Entity Recognition (NER) plays an essential role in a variety of NLP applications such as text understanding [99], information retrieval [38], automatic text summarization [71], question answering [67], machine translation [13] and knowledge base construction [33].

We will review NER task with a general approach so different techniques and methods are introduced leading to a more general overview, but it is important to notice that core experimental work is focused on biomedical NER. Name entity recognition task in biomedical domain focuses in entities such as proteins, species, diseases, chemical or mutations. This particular domain has special difficulties given the heterogeneity and polysemy

in some entities such as genes, symptoms and diseases. Biomedical NER is a fundamental task in larger scale applications such as network biology analysis [104], gene prioritization [9], drug repositories [89] or curated databases creation [52]. As it will be discuss in Section 3, biomedical NER has also its own set of resources such as Corpus and NER systems.

Having complex biomedical tools based on powerful NER systems could derive in better healthcare systems improving living standards around the globe. It could also help to share biomedical knowledge between different language and culture research communities accelerating research in biomedical domains. NER task has also technical motivations, designing a state of art NER system could help to improve our knowledge about Deep Learning architectures. One point of technical interest is how to take the most from the system when we are in a low resources environment, i.e there is not a large amount of training data, and we need to choose which text segments use for training.

1.2 Objectives

In this work, we want to put together several general techniques for NER task in order to have a global overview of how to accomplish a NER task successfully and how this techniques have evolved over the years. In this revision, we also want to expose some relevant associations, evaluation campaigns, corpus and resources for NER task, not only in general but also in biomedical domain, in order to give also practical information on how to face NER task and recognize the community efforts to improve overall NER systems performance.

We will design a NER system based on the popular Deep Transformer, BERT, to accomplish NER task in a cross-domain and cross-lingual setting using eHealth Knowledge Discovery Challengee at IberLEF 2021 evaluation campaign [93] available corpus. Then, we will study how this system performs when we select certain sentences in the training corpus. We would like to discover if there are some sentences that provide more information than others in the corpus, and if we would be able to choose them from a bigger corpus using simple features extracted from the sentences. This would help to reduce training costs and to prioritize certain sentences when deciding which ones should be annotated by expert teams.

Finally, we will evaluate the system of the previous experiment to accomplish the

eHealth-KD 2021 task which focuses in a cross-domain, multi-lingual and low resources environment given its small amount of training data. This setting presents a significant challenge to existing state-of-the-art methods, which often rely on large amounts of training data.

1.3 Thesis outline

This work is structured as follows:

- **Chapter 1.** In this chapter, we describe the problem that will be covered in this work, as well as motivations and objectives.
- **Chapter 2.** A historical overview of NER task is presented. We introduce several techniques and works with its main conclusions. This overview has been made with a general purpose but some examples of NER in biomedical dominal, multidomain and multilingual can be seen.
- **Chapter 3.** This chapter puts together many evaluation campaigns, corpus and resources. Also, characteristics of this resources are explained. It tries to be a repository for those who face this problem for the fist time.
- **Chapter 4.** In this chapter, hypothesis is presented as well as a proposed system description. We will discuss the ideas we want to prove and the system that we will use to do so. We cover our system corpus pre-processing, model and corpus post-processing steps.
- **Chapter 5.** Experiments that have taken place are explained in detail. We explain technical configurations and present analysis and calculations that justify some of our design decisions. Results and its discussion are also presented in this chapter.
- **Chapter 6.** We cover the conclusions of this work as well as possible further research lines.

Chapter 2

Name entity recognition. A historical overview

The first approach to NER was based on handcrafted rules that allowed to recognize some words from an specific field. In [77], a system capable of extracting and recognize company names was presented. It was based on heuristics and handcrafted rules.

More advanced techniques based on supervised, semi-Supervised, unsupervised learning (Section 2.1) and deep learning (Section 2.2) will be described in further Sections.

2.1 Machine Learning

Machine Learning (ML) is a field of artificial intelligence focused on developing methods that are able to learn patterns from historical data in order to improve performance on some set of tasks. In general terms, machine learning algorithms build models that learn from past experiences (which in this case is previous data) to improve future performance. Machine Learning has been a computer science research field since the 60s, anyway it was not until the 90s when development in computational capabilities and data gathering made this research domain speed up. Nowadays, Machine Learning is a very popular field in research and private companies and its applications are broadly extended in society. In this Section, different Machine Learning approaches for NER will be introduced.

2.1.1 Supervised Learning

Supervised learning is based on labeled data for training. Normally, corpus with human labeled entities are used. These corpora are known as Golden Standard. For models to have a good performance it results fundamental to extract good features from corpus that allow models to learn patterns with rich information. These feature extraction techniques will be discuss in Section 2.1.4.

Several different approaches have been published in the last years. These works will focus on those based on Hiden Markov Models (HMM), Maximum Entropy Models, Support Vector Machines (SVM) and Conditional Random Fields (CRF).

Hidden Markov Models (HMM)

Hidden Markov Model are statistical models in which the modeled system is assumed to be a Marvok process. Markov processes [43] describe systems that have limited memory of their past. In case we are processing text sequentially, it would represent that given one sequence of words the next word would only be influenced by the last one not taking into account the rest of the sequence. To define it more precisely, a random process $\{X(t), t \in T\}$ is called a fist-order Markov process if for any $t_0 < t_1 < ... < t_n$ the conditional probability of $X(t_n)$ for given values of $X(t_0), X(t_1), ..., X(t_{n-1})$ depends only on $X(t_{n-1})$, that is:

$$\begin{aligned} P[X(t_n) \leq x_n | X(t_{n-1}) \leq x_{n-1}, X(t_{n-2}) \leq x_{n-2}, ..., X(t_{n-0}) \leq x_{n-0}] \\ = P[X(t_n) \leq x_n | X(t_{n-1}) \leq x_{n-1}] \end{aligned} \tag{2.1}$$

Now we will introduce some works that design NER systems based on Hidden Markov Models.

In [15] **Bikel, et al.** propose a system called *Identifier* able to add a classification label, including *Not Available* tag for those non classifiable words, to every entity in a sentence. The system tries to find the most likely entity labels sequence for a given words sequence:

$$maxP_r(NC|W); P_r(NC|W) = \frac{P_r(W|NC)}{P_r(W)} \tag{2.2}$$

Where W represents the word sequence and NC, the name class sequence. As can be observed from the equation, this is a generative model that allows us to predict the next most probable word and name class. Viterbi algorithm [35] is used for $P_r(NC|W)$ optimization in all possible NC space.

Identifier reaches 94.4% accuaracy in MUC-6 task on data and collections from Wall Street Journal and 90% in MET-1 a mix of Spanish corpus.

Zhou [102] improved *Identifier* using mutual information. Given a token (words) sequence $G_1^n = g_1 g_2 ... g_n$ the objective is finding the best sequence of labels (tags) $T_1^n = t_1 t_2 ... t_n$ that maximizes:

$$P_r(T_1^n|G_1^n) = logP_r(T_1^n) + log\frac{P_r(T_1^n, G_1^n)}{P_r(T_1^n)P_r(G_1^n)} \tag{2.3}$$

This approach is able to extract NE in noisy environments. This system reaches 96.6% accuracy on MUC-6 and 94.1% accuracy on MUC-7.

Maximun Entropy models (MaxEnt)

Maximum entropy is a method of statistical modeling which turn on the notion of *futures*, *histories* and *features* [17] . Futures are defined as the possible outputs of the model. A maximum entropy solution to NER, or any other similar problem, allows the computation of $p(f|h)$ for any f from the space of possible futures, F, for every h from the space of possible histories, H. History is all of the conditioning data which enables to assign probabilities to the space of futures. In the name entity task, we could reformulate the maximum entropy problem in terms of finding the probability of f associated with the token at index t in the test corpus as:

$$p(f|h_t) = p(f|information\,of\,token\,t\,extracted\,from\,test\,corpus) \tag{2.4}$$

The computation of $p(f|h)$ in M.E. is dependent on a set of *features* that help making a prediction about the future. This features are binary functions of the history and future.

In [17] a MaxEnt system is proposed, its name is Max Entropy Name Entity (MENE). This system learns the weights for discriminating features for classification given a set of precalculated features and training data. The main objective is maximizing entropy so the system is able to generalize as much as possible.

Probability of a label, f, given the historical data, h, is:

$$P(f|h) = \frac{\prod_t \Lambda_t^{g_t(h,t)}}{Z_\lambda(h)} \tag{2.5}$$

with Z:

$$Z_\lambda(h) = \sum_f \prod_t \Lambda_t^{g_t(h,f)} \tag{2.6}$$

where λ represents each feature parameter. Maximize entropy, Z, ensures that for all g_t the expected value of g_t will be equal to the empirical expectation of g_t in the training corpus. In the last step Viterbi algorithm is used to find the highest probability path trough the series of conditioned probabilities which produces the tag sequence.

MENE [17] system got data from different sources of text and generated text features such as lexical features, section features, external system output, consistency and reference resolution. Given a set of tags, 29 tags of MUC-7, this system is able to make computation of $p(f|h)\forall f$ of features space, which is the possible output of the 29 tags in this case, and $\forall h$ from the space of histories (H), where a history is all the conditional data needed to maximize entropy and make decisions about the future. This system had results of 88.8% accuracy in MUC-7.

Another approach of ME models is Curran's ME Tagger [25] where the use of softmax function is proposed to formulate probability:

$$P(y|x) = \frac{1}{Z(x)} exp(\sum_{i=1}^{n} \lambda_i f_i(x, y)) \tag{2.7}$$

where y is the tag, x is the context and f_i is the feature associated to λ_i. Then the probability of $y_1, ..., y_n$ tags of $w_i, ..., w_n$ word sequence is $P(y_i...y_n|w_i...w_n) = \prod_{i=1} Pr(y_i|x_i)$. This system got good results in accuracy for different languages from CoNLL-2003 and CoNLL-2002 data sets: 84,89% for English and 68,48% for German.

Suport Vector Machines (SVM)

Suport Vector Machines (SVM) were introduced by [23] based on the idea of finding a linear hyper plane that separates positive examples from negative ones by large margin, i.e. large distance in features hyper plane.

In [62], the problem was tackled as a binary decision problem and one classifier per tag was trained. An important work of feature generation was made introducing 258 orthography and punctuation features and 1000 language-relate binary features. Also, a 7 words size windows is used to take information of the context, and this makes the number of features arise to 8806. For CoNLL-2002 data accuracy was 60.97% for Spanish and 59,52% for Dutch.

Conditional Random Field (CRF)

Conditional Random Field (CRF) was introduced by [48] as a statistical modeling tool for pattern recognition and ML using structure prediction.

In [61], a feature induction method for CRF in NE is proposed. Let $o =< o_1, o_2, ..., o_T >$ be some observed data, text in our case. Let S be a set of finite state machine, FSM states, $S =< s_1, ..., s_T >$, each with label L. By Hammersley Clifford theorem [39], CRF define the probability of a state sequence given an input sequence to be:

$$P(S|O) = \frac{1}{Z} exp(\sum_{t=1}^{T} \lambda_k f_k(s_{t-1}, s_t, o, t)) \tag{2.8}$$

Where Z is the normalization factor, $f_k(s_{t-1}, s_t, o, t)$ is an arbitrary feature function and λ_k is the learned weight of each feature function. Using dynamic programming, state transition between two CRF states can be efficiently calculated:

$$\alpha_{t+1}(s) = \sum_{s} \alpha_t(s') exp(\sum_{k} \lambda k f_k(s', s, o, t)) \tag{2.9}$$

where $\alpha_t(s)$ is the *unnormalized probability* of arriving state s_i given the observations $< o_1, o_2, ..., o_T >$. $\alpha_0(s)$ is the set of probability of starting in each state s. Now Z_o is calculated as $\sum_s \alpha_T(s)$ and Viterbi algorithm is used to find the most likely state sequence given the observation sequence. Results on CoNLL 2003 give an accuracy of 84.04% for English and 68.99% for German.

2.1.2 Semi-supervised Learning

Semi-supervised Learning algorithms typically start with small amount of seed data and create more hypothesis using large unlabeled corpus. Semi-supervised learning needs less human effort than supervised learning and this makes it very interesting in theory and practice [105]. Anyways, semi-supervised systems are less frequent than others.

One of the first approaches for NER task was proposed in [18], where regular expressions system based on lexical features was implemented to generate a list of book titles paired with book authors. For example, given {Isaac Asimov, The Robots of Dawn} leads to a regular expression like $[A-z][A-za-z.,\&]^{5,3}[A-Za-z.]$ that can describe other books and authors found in the web.

Later, in [21], a complete text is parsed by searching for a candidate NE pattern. For example, proper none followed by a noun phrase in apposition. Patterns were kept in pairs (spelling, context), for example (New York, location). Then, contextual rules are inferred by analyzing all the accumulated context information. In a further work [95], new spelling rules are learned from this context information.

Another approach based on AdaBoost classifier is proposed in [19]. Boosting algorithm is formed by trees. Each tree learns sequentially by presenting the decision tree a weighting over the examples which depend on the previous learned trees. Here BIO labeling scheme is used to define NER problem. Each word sequence is considered as input and has to be labeled with one of the BIO labels: Beginning of the NE (B-), Inside the NE (I) or Outside (O-). Orthographic and semantic features are evaluated over a shifting window allowing a relational representation of examples via many simple binary propositional features. The boosted decision trees construct conjunctions of such binary features, allowing the boosting classifier to work with complex and expressive rules. Three classifiers are trained for each kind of tag and they process the sentence from left to right, selecting for each word the tag with maximum confidence that is coherent with the current solution. The semi-supervised training is forced as only 40% of the NE example in training set is used and then knowledge is expanded to the whole data set using also external examples. This system achieves 79.9% accuracy in CoNLL 2002 shared task in Spanish.

2.1.3 Unsupervised Learning

Unsupervised learning algorithms learn to identify patterns in data sets where data points are not labeled nor classified. Usually this algorithms allows to classify data into groups within the data set without any human guidance [32].

Several unsupervised methods have been proposed for NER task. One of the first examples is [12]. This work studies the task of labeling and input a word with an appropriate NE type taken from WordNet. The approach is to assign a topic signature to each Wordnet synset by merely listing words that frequently co-occur.

Work [34] apply [40] method to find potential hypernyms of sequence of capitalized words appearing in a document. For example, when X is capitalized the phrase *such as X* is searched on the web and the noun that immediately precede the query can be chosen as the hypermedia of X.

KNOWITALL is proposed in [33], an information extractor from the web in an unsupervised and open-ended manner. It uses eight independent extraction patterns. For example, pattern *NP1 such as NPlist2* indicates that NP1 is in the list of NP2. Then this statement is checked testing the probability of the candidate using mutual information (PMI) computed using large web text as corpus. Based on PMI score, KNOWITALL associates probability with every fact it extracts.

An unsupervised NER system across languages can be seen in [69]. This system generates seed candidates through local, cross language edit likelihood and then, bootstraps to make broad predictions across two languages. It is completely unsupervised with no manual labels. It is only based on parallel text that does not need to be easily alignable. It gets results of F1=0.85 on parallel corpus of English and Haitian Kreyol used in 2010 shared task for the workshop on Machine Learning translation.

2.1.4 Feature extraction and engineering

All the different machine learning methods of the previous section are based on features to train and learn. As a more formal definition, features are descriptors or characteristic attributes of words designed for algorithmic consumption. In a traditional approach NER task was solved as a ruled based system based on this features. The natural evolution are ML systems where rules are learned automatically during the training process.

Given a corpus, we distinguish three different features that can be extracted. Those are word-level features, list look-up and document, corpus-level features [70].

Main word-level features are:

- **Digit pattern**. Number recognition helps point years, amounts, etc.
- **Common word ending**. This morphological feature is useful relating words of similar classes.
- **Case**. Capitalized words can denote entities as well as upper words could be acronyms.
- **Punctuation**. The location of periods, apostrophe gives helpful information about relation between words in the corpus.
- **Characters**. Paying attention to specific characters such as possessive mark or greek letters give information about text structure.

- **Part of speech**. Part of speech information such as proper noun, verb, noun, adjective is basic for any NLP task.

- **Functions**. More complex features such as those extracted from n-grans, alpha and non-alpha functions, pattern summarization or simpler such as word length can also be used.

Lists, gazetteers, lexicon or dictionaries are privileged features in NER tasks. If a word is included in a list, then the probability of that word to be a name entity is high. For example, [64] demonstrated that using dictionary look-up techniques allows disambiguation of 80% of those words in ambiguous positions. Other typical lists include those with organizations endings that allow recognize organization names in a text. Also, look-up techniques often include some flexibility in the matching condition like inflected terms (technology, technologies), fuzzy-matching (spell variation) and more complex systems like *soundex* which normalizes words to its phonetics. Using *soundex*, Lewinskey would be equivalent to Lewinsky.

The last type of feature engineering techniques are those extracted from documents and corpus. These features are defined both over document content and document structure. One example of features are multiple occurrences, in [84] words that appeared upper and lower case were identified so they are hypothesized to be common nouns. Another example are occurrences of a given word or word sequence referring to a given entity within a document. Exploiting the context of every occurrence some features can be derived. For example for the sentence *MacDonald was the first partner*, next time that references to the first partner are shown in text a relation to MacDonald can be establish. There is also some meta information about the texts that can be used directly to find name entities. For example, an email header for names or news first lines for location names. Lastly, statistics for multi-words units can be developed for feature extracting. For example, [26] proposed feature functions for multi-word units that can be used as a threshold for NE detection. For example, threshold on the presence of rare and long lowercase words in entities, where rarity can be computed as the relative frequency in the corpus. Only multi-words units that do not contains relatively long size words are considered as NE candidate.

2.1.5 Feature Selection

Feature selection, as a dimensionalitiy reduction technique, tries to choose a subset from the original features by removing irrelevant, redundant or noisy features [63]. As we have seen in previous section, Machine Learning algorithms for NLP in general and NER in particular are heavily influenced by the information given by text features (Section 2.1.1) that have been extracted with feature extraction methods (Section 2.1.4). However, using, for example, systematical statistical feature extraction, such as frequency information for each vocabulary word, could lead to a huge number of features. This increase on dimensionality could decrease the learning performance, this is known in data science community as *the curse of dimensionality.*

We will visit some general techniques of feature selection that could help us escape from the curse of dimensionality. On the one hand, based on the availability of label information, different methods can be classified in supervised methods, semi-supervised methods and unsupervised methods. On the other hand, based on the strategies of searching can be classified in filter methods, wrapper methods and embedded methods. We will visit some main feature selection algorithms.

- **Supervised.** Label information is used to choose the most important features. In [92], a method based on spectral properties of the Laplacian of the features' measurement matrix is proposed. Then, the feature selection process is based on a continuous ranking of the features defined by a least-squares optimization process.

- **Semi-supervised.** Available label information is used to choose the most important features. A semi-supervised method is proposed in [100]. It is a selection algorithm based on spectral analysis. The algorithm exploits both a minority of labeled data and a majority of unlabeled data through a regularization framework based on clustering.

- **Unsupervised.** Only patterns extracted from the data itself are used to choose the most important features. In [59], PCA method is described. This method calculates and chooses the components that maximize data variance first. Nowadays, PCA is one of the most extended methods.

- **Wrapper.** Wrapper methods use the learning algorithm that is going to be applied after feature selection to evaluate the features. For example, algorithms like regres-

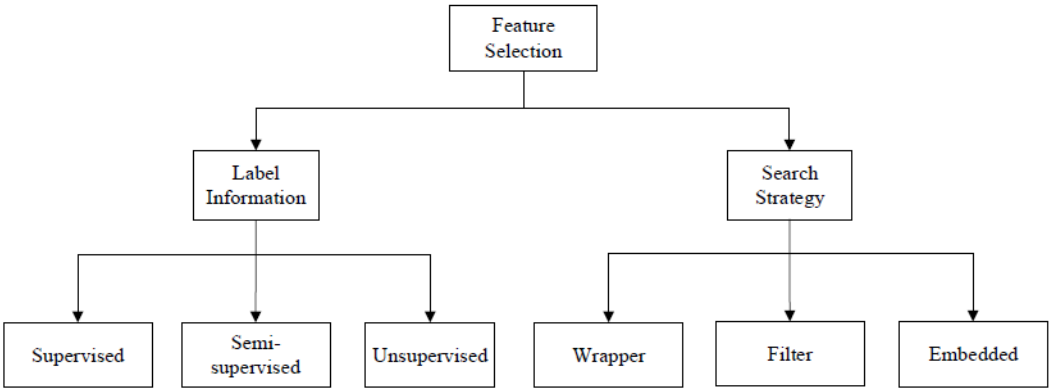

Figure 2.1: Schema of different feature selection techniques [63].

sions or decision trees give information of the feature importance and this information can be use to apply the feature selection.

- **Filter.** Filter methods select the most discriminative features through the character of data. First, all features are ordered given certain criteria and then those with highest information are chosen.
- **Embedded.** Embedded models perform feature selection in the model construction process.

2.2 Deep Learning

First application of Deep Learning (DL) based system for NER was proposed in [22]. These systems, in contrast with those ruled based and machine learning based systems, have minimal feature engineering and have been flourishing during the last years.

The reason why Deep Learning is increasing its influence is related to the capabilities of its architecture. Multiple processing layers can learn representations of data with multiple levels of abstraction. Layers are artificial neural networks which consists of the forward pass and backward pass. The forward pass computes a weight sum of their inputs from the previous layer and pass the result through a non-linear function. The backward pass is to compute the gradient of an objective function with respect to the weights of a multilayer stack of modules via chain rule of derivatives. The key advantage is the capability of representation learning and the semantic composition empowered by both the vector

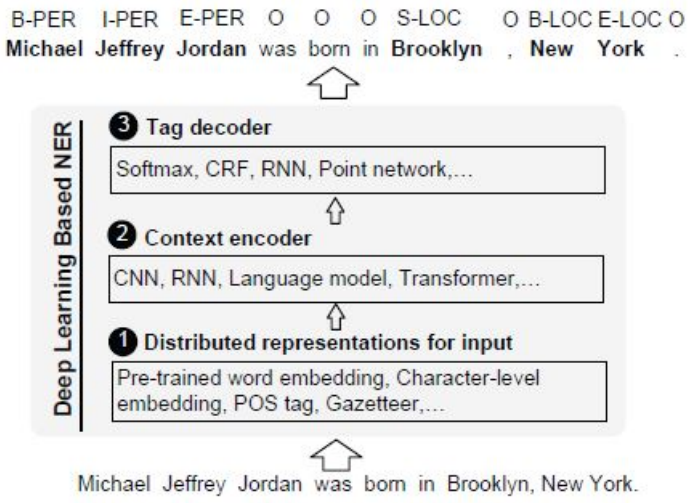

Figure 2.2: Taxonomy for a Deep Learning approach for NER task [53]

representation and neural processing. This allows a machine to be fed with raw data and to automatically discover latent representations needed for classification or detection [50].

There are three core strengths of using deep learning techniques to NER. First, the deep learning non linear transformation can generate non linear mappings between inputs and outputs and this approach is more similar to real world problems. Second, time saving on designing and extracting features is significant. Lastly, deep learning for NER can be trained in an end-to-end paradigm. This means that only by gradient descent optimization we can get the most from a NER system and given this simpler methodology accomplish more complex tasks.

Actually, as it has been visited in Section 2.1, it is useful to remember that traditional ML systems for NER were focused in one domain and that cross-domain systems didn't show great results until deep learning techniques such as transfer learning were broadly adopted. The authors of [75] trained a NER system on MUC-6 journalistic data sets and applied it on a technical domain corpus. A drop in performance around 20%-40% on precision and recall was observed.

The basic steps in deep learning NER systems are **Distributed Representation for Input**, Section 2.2.1, **Context Encoder**, Section 2.2.2, **Tag Decoder** , Section 2.2.3. Examples of these steps can be seen in Figure 2.2

2.2.1 Distributed representation for input

The first step in Deep Learning is transforming the input into a distributed representation so it can be processed by the deep learning architecture. The simplest option is **one-hot vector representation**. This representation generates a vocabulary length vector in which each position represents a word in the vocabulary. Then, a word is represented with a one in its vector position. Using this representation two words are orthogonal.

The next step in complexity consists on using **distributed representations**, in which words are represented in low dimensional real-valued dense vectors, where each dimension represents a latent feature. Automatically learned from text, this representation captures semantic and syntactic properties of words, which do not explicitly present the input of NER.

Two main distributed representation can be defined depending on the level of information that is used. These are word-level representations and character-level representations.

Distributed word-level representation

In [101], a pre-trained word-level representation is employed over large collections of text trough unsupervised algorithm such as [65] continuous bag of words (CBOW) and continuous skip-grams. In [82] the importance of pre-trained representation such as Google Word2Vec [65], StandFord Glove [72], Facebook fastText [16] [44] or SENNA [22] is studied and demonstrated.

Bio-NER, a biomedical NER model based on deep learning network architecture is presented in [96]. It is trained on PubMed database using skip-gram model. The dictionary contains 205924 words in 600 dimensional vectors.

Distributed character-level representation

In [47], a character-based word representation approach is incorporated. They learned from and end-to-end neural network model. Exploiting explicit sub-word information such as suffix and prefix has been found useful. Also, it naturally handles out-of-vocabulary words because it learns from specific characters. There are two widely used architectures: Convolutional Neural Networks, CNN (Figure 2.3) and Recurrent Neural Networks, RNN (Figure 2.4).

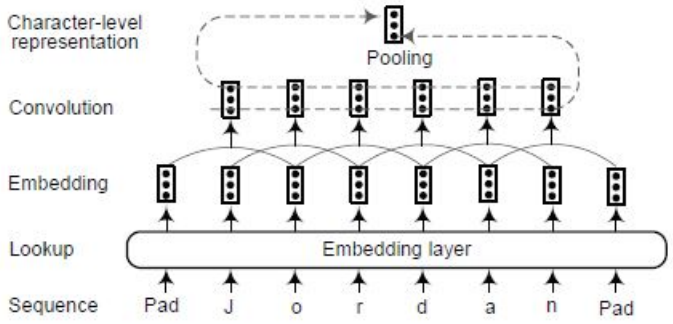

Figure 2.3: CNN architecture schema [53].

Convolutional Neural Networks, CNN

The first introduction of modern CNN architecture was made in 1998 [51] for a NLP approach but it wasn't until 2012 when ImageNet Large Scale Visual Recognition Challenge was won with a CNN architecture. It was at this point that CNN started to be one of the reference architecture for artificial vision first and NLP later. An schema of CNN architecture for NER can be seen in Figure 2.3.

In [58], it is possible to see an example of utilization of a CNN character-level representation. Then the character representation vector is concatenated with the word embedding before feeding into a RNN context encoder.

In [74], ELMo word representation is proposed, which is computed on top of two-layer bidirectional language model with character convolutions.

A neural reranking model for NER where a convolutional layer with a fixed window-size is used on top of a character embedding layer is proposed in [94].

Recurrent Neural Networks, RNN

First introduction to RNN was in 1995 [91], when its dynamics for a simple NLP task were studied. Although recursion works well for short term information, it performs poorly for long term information. For this reason, RNN evolution, Long Short Term Memory [41] architecture, are more popular nowadays. An schema of RNN architecture can be seen in Figure 2.4.

Two typical choices of RNN units are Long Short Term Memory (LSTM) [41] and Gate Recurrent Unit (GRU) [14]. In [47], charNER is introduced, a character-level tagger for

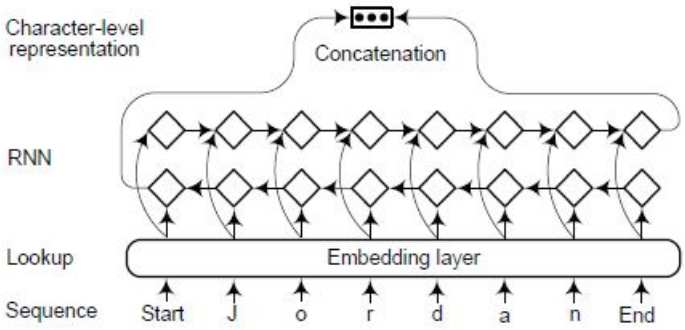

Figure 2.4: RNN architecture schema [53].

language independent NER. CharNER considers a sentence as a sequence of characters and utilizes LSTMs to extract character-level representations. It outputs character level tags and word level tags arise from character level tags. Results are better than taking words as basic units.

In [79], a system that combined character-level representations with word embeddings using gating mechanism (GRU) is proposed. Model dynamically decides how much information use from character or word level.

Contextual string embedding is proposed in [11]. It uses character-level neural language model to generate a contextualized embedding for a string of characters in a sequential context. Embeddings are contextualized, meaning that a word has different embedding depending on the surrounding words.

Hybrid representation

Although both RNN and CNN approaches are powerful, more information can be incorporated to the final representation of words (character or word level). For example: gazetteers, lexical similarity, linguistic dependency and visual features.

The authors of [42] include features extracted from gazetteers that boost tagging accuracy in spelling features, context features an word embedding systems.

One example of lexical similarity information incorporation is [90]. It proposes a CRF-based neural system for recognizing and normalizing disease names. It employs word embeddings, POS-tags, chunking and word shape features.

As an example of linguistic dependencies information application, the authors of [10] propose a CNN system that is able to capture orthographic features and word shapes at

character level. For syntactical and contextual information at word level, e.g POS tags and word embeddings, the model implements a LSTM architecture.

A multimodel NER system for noisy user-generated data like tweets and snapchat captions is proposed in [68] . Word and character embeddings and visual features are merged with modalitiy attention.

The popular language representation BERT (Bidirectional Encoder Representation from Transformers) is proposed in [29]. It uses masked language models to enable the use of pre-trained deep bidirectional representation. For a given token, its input representation is compiled by summing the corresponding position, segment and token embeddings. This pre-trained language model requires large corpora and incorporates auxiliary embeddings so it can be considered a hybrid representation.

2.2.2 Context Encoder

Once we have an input representation, it is time to try to capture the context dependencies using context encoders. These context encoders can be relatively simple DL architectures or more complex DL language models. In this section, we will introduce the most widely-used architectures: convolutional neural networks, recurrent neural networks, recursive neural networks and deep transformers.

For a first approach to context encoders, it could be confusing to address the differences between them and word representations such as word embeddings. We think it is useful to make this concept clear before getting into more detail about context encoders. We would remark that the main difference is that, given a certain word, word embeddings (like Word2Vec or GloVe) always give the same representation, usually in a vector form, while context encoders (like popular Deep Transformers like BERT or GPT-3) will give different representations depending on the words that surround it in the sentence.

Convolutional Neural Networks, CNN

The first time CNN were applied as context encoders was in [22], were a sentence approach network was proposed. In this system, every word is tagged with the consideration of the whole sentence.

Each word is an element of an N-dimensional vector after the stage of input representation. Then a convolutional layer is used to produce local features around the number of

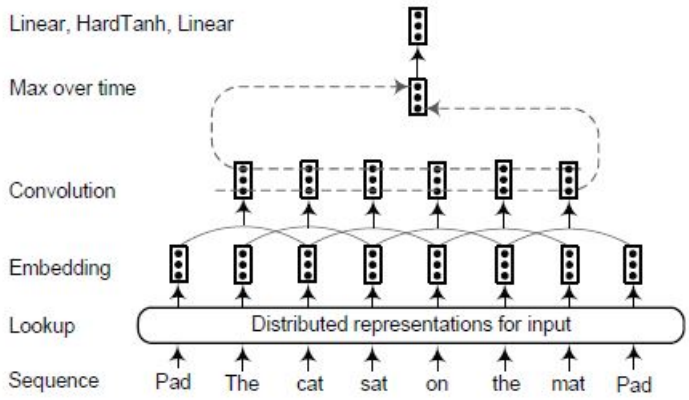

Figure 2.5: CNN architecture for Context Encoder [53].

words in the sentence. A global feature vector is constructed by combining local feature vectors extracted by convolutional layers. The dimension of this vector is fixed, independent of the sentence length, in order to apply subsequent standard related layers. To extract global features, two approaches are widely used: **Max** or **averaging** operation over the position in the sequence. Finally, these fixed-sized global features are fed into a tag decoder to compute distribution scores for all possible tags for the words in the network input. Architecture schema can be seen in Figure 2.5.

One example of CNN application can be seen in [103]. In this work it is shown that RNN are focused on last words more than in former words but important words can appear anywhere. BLTSM_RE system is proposed, BLTSM is used to capture long-term dependencies and obtain the whole representation. Then a CNN is utilized to learn a high level representation which is then fed into a sigmoid classifier. Then, the whole sentence representation (BLSTM) and the relation representation (sigmoid) are fed to another RNN-LSTM to predict entities and relationships getting BLSTM_RE.

Recurrent Neural Networks, RNN

Recurrent Neural Networks, RNN, with its variants GRU (gated recurrent unit) and LSTM (long short term memory) have demonstrated remarkable achievements in modeling sequential data. Bidirectional RNN make an efficient use of past information (via forward states) and future information (via backward states). This architecture can be seen in Figure 2.6 . This results are very useful when processing language and it is why bidirectional RNN have become an standard of deep context-dependent representations of text [83] [58].

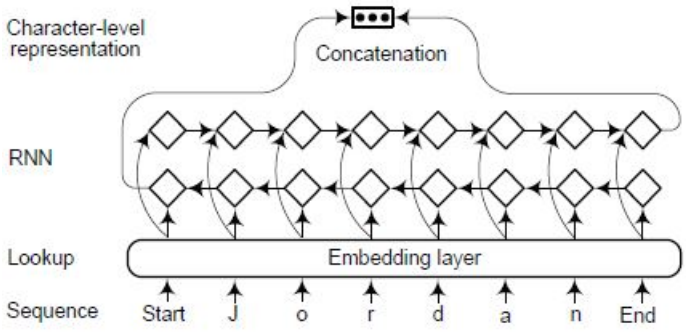

Figure 2.6: RNN architecture for context encoder [53].

One of the first works that uses a bi-LSTM CRF achitecture to sequence tagging tasks (POS, chunking, and NER) is [42]. Following this work some other have followed its path [56] [85] [49].

In [94], it is presented an example of employment of deep GRUs on word and character level to encode morphology and context information. They extended their model to cross-lingual and multi-task joint train by sharing architecture and parameters.

A modification of standard LSTM-based sequence labeling model, so it is able to handle nested name entity recognition as well, is proposed in [45].

Recursive Neural Network

Recursive Neural Networks are non-linear adaptive models able to learn deep structure information by transforming a given structure into a topological order [54]. As NE are strongly related to linguistic constituents e.g. nouns, verb phrases... this methodology results very useful to detect deep morphological and syntactical relationships. An architecture schema of Recursive Neural Network can be seen in Figure 2.7.

In [54], a system able to classify every node in a constituency structure for NER is proposed. This model calculates recursively hidden state vectors of every node and classifies each node by these hidden vectors.

Neural Language Models

Language models are a family of models describing the generations of sequences. Given a token sequence (t_1, t_2, ..., t_N), a forward language model computes the probability of the sequence by modeling the probability of token t_k given the history (t_1, ..., t_{k-1})

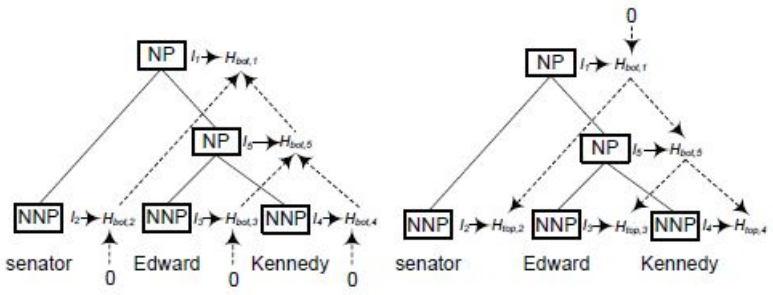

Figure 2.7: Bidirectional Recursive Neural Network architecture for NER [53].

$$p(t_1, t_2, ..., t_N) = \prod_{k=1}^{N} p(t_k | t_1, t_2, ..., t_{k-1}) \tag{2.10}$$

A backward language model is similar to a forward language model, it predicts previous token given its future context.

$$p(t_1, t_2, ..., t_N) = \prod_{k=1}^{N} p(t_k | t_{k+1}, t_{k+2}, ..., t_N) \tag{2.11}$$

The probability of token t_k can be calculated by output of recurrent neural networks. For each position k, a backward and forward model can be applied and then, combining both representations, the final language model embedding for token t_k is computed.

In [78], it is proposed a framework with a double objective: NER and prediction of surrounding words for each word in the data set. At each step, the network is optimized to predict the previous token, the current tag, and the next token in the same sequence. This double objective enriches the feature representation that is used for NER.

TagLM, a language model augmented sequence tagger is proposed in [73]. This tagger consider pre-trained word embeddings and bidirectional language models embedding for each token in input sentence.

ELMo representation is proposed in [74]. This representation is computed on top of two-layer bidirectional language models with character convolutions. It models semantics and syntax and can be used across languages.

Deep transformers

Labeling models are based on recurrent networks which consist of encoders and decoders.

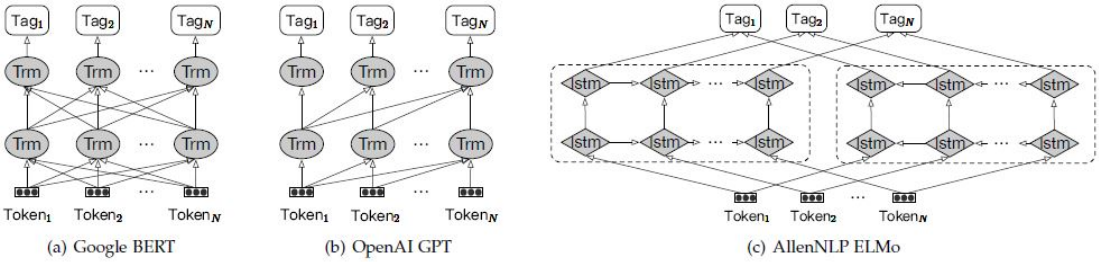

Figure 2.8: Differences in Deep Transformers architectures[53]: Google BERT In [29], OpenAI GPT [76] and AllenNLP ELMO [74] .

In [86], a transformer which dispenses with recurrence and convolutions entirely is proposed. Transformers utilized self-attention and point-wise, fully connected layers to build basics blocks for encoder and decoders. It is shown that transformers perform better and require less time to be trained.

A generative pre-trained transformer (GPT) for language understanding is proposed in [76]. GPT has two stages. First, it uses a language modeling objective with transformers on unlabeled data to learn initial parameters. Then, it adapts the parameters to the target task using a supervised objective resulting minimal changes. It is a left-to-right architecture.

Bidirectional Encoder Representations from Transformers (BERT) is proposed in [29]. BERT's key technical innovation is applying the bidirectional training of Transformer to language modelling. It is shown that a bidirectionally trained language model can have a deeper sense of language context than single-direction language models.

These pre-trained language model embeddings using transformers are the new paradigm of NER. Differences in its architectures can be seen in Figure 2.8. This models can replace traditional embeddings such as Google Word2Vec or Standford Glove. However, both traditional and new approaches combined can lead to better performances as shown in [57].

Also, this models can be further fine-tuned with one additional output layer for a wide range of task including NER and chunking. In [55], it is demonstrated that machine reading comprehension (MRC) can be approached by fine-tuning BERT model.

2.2.3 Tag Decoders

Decoder is the final stage in NER deep learning models. It takes as input context-dependent representation, and produces a sequence of tags corresponding to the input sequence. There

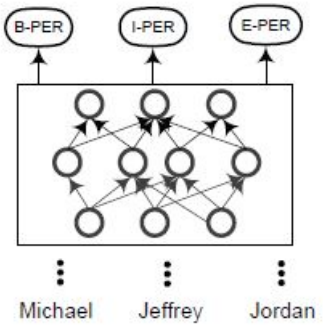

Figure 2.9: Multilayer Perception and Softmax architecture for tag decoding [53].

are four main architectures of tag decoder: Multilayer Perception and softmax, conditional random fields, recurrent neural network and pointer networks.

Multilayer Perception and Softmax

The labeling task is approached as a multi-class classification problem. Tag for each word is predicted using its context-dependent representation, but without using its neighbor representation. Some examples can be found in [29] and [24]. Decoder system can be seen in Figure 2.9.

Conditional Random Fields

Conditional Random Fields technique was previously introduced in Section 2.1 and presented in [48]. The typical use of CRF on NER task is on top of a LSTM layer [42] and on top of a CNN layer [96].

CRF are the most commonly used decoders. However, they cannot make full use of segment-level information because segment properties cannot be represented at word level representation. In [103] and [97], a semi-markov modification to CRF is proposed, so model segments is also possible. CRF decoder architecture can be seen in Figure 2.10.

Recurrent Neural Network

We will describe how Recurrent Neutral Network works for decoding. The architecture starts processing the information in the Go symbol in Figure 2.11. In this first step, this signal is provided as y_1 to RNN decoder. Then, at each step i, RNN decoder computes current decoder hidden state h_{i+1}^{Dec} in terms of previous step tag y_i, previous step decoder

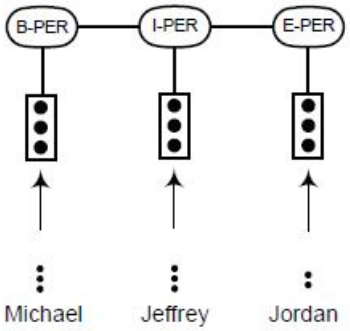

Figure 2.10: Conditional Random Field architecture for tag decoding [53].

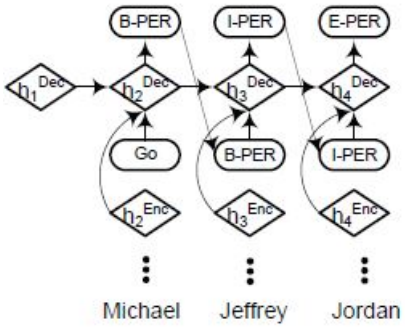

Figure 2.11: RNN architecture for tag decoding [53].

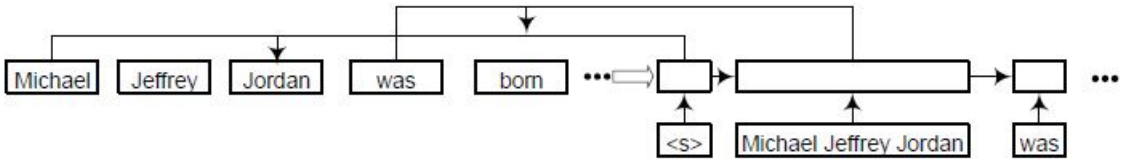

Figure 2.12: Pointer Networks decoder architecture [53].

hidden state h_i^{Dec} and current step encoder hidden state h_{i+1}^{Dec}. The current output tag y_{i+1} is decoded using softmax and it is further fed as an input of the next step. Finally, a tag sequence over all time steps is obtained. In [82], this architecture is used to get better and also faster results than CRF decoder based architectures.

Pointer Networks

In general terms, pointer networks apply RNNs to learn conditional probability of an output sequence, given a sequence of discrete tokens corresponding to the positions in an input sequence [87]. It is able to represent variable lengths dictionaries by using a *softmax* distribution as a *pointer*. The first example of pointer networks for NER is presented in [98].

In the Figure 2.12, the pointer network first identifies a chunk or segment and labels it. Later it points to the next segment until all the sentence is analyzed.

2.3 Historical Review Conclusions

As we have seen in the historical review, there are many different and diverse approaches to NER task. Importance of NER task in NLP domain makes it an interesting field and many efforts to design and adapt models and architectures have been made.

Thank to the increase of computational power, such as GPU infrastructure, Deep Learning approaches are growing in number and forms. This techniques are more architecture oriented than first NER techniques, such as rule based or simple ML models based, which were more data understanding oriented. Anyways, these DL solutions perform better than almost any other previous approach.

Although many architectures and models have been introduced, finding good quality and entity annotated corpus is not that frequent. This states a challenge to find corpus, annotations and evaluation standards that allow research community to easily compare

techniques and walk in a more straight way to find the best solutions to different tasks including NER.

This work will show the effectiveness in multidomain, multilingual and low resources environment NER task, of a Deep Transfomer architecture (Section 2.2.2) combined with the simplest input representation possible, as it is one-hot vector representation (Section 2.2.1), and the default BERT tag decoder. We will see how this simple approach behaves when we choose which sentences to use for training in a low resource environment. This could help us identify patterns in the corpus that allow us optimize training. Also, regarding to the work of creating golden source corpora (Sections 2.1.1 and 3.2.1), this knowledge can be apply to annotate first those sentences that will have a better impact in systems training. We will also see how this simple approach leads to a campaign level performance.

Chapter 3

Resources and evaluation campaigns for NER

3.1 Associations

In this section, we want to sum up several associations and evaluation campaigns that make efforts to push NER techniques a step further, proposing contest and tasks in various domains.

3.1.1 CLEF

CLEF [2] is the acronym for Conference and Labs of the Evaluation Forum. This european self-organized body was founded in 2000 and promotes research, innovation and development of information access systems with an emphasis on multilingual and multimodal information with various levels of structure.

CLEF initiatives are structured in two main parts: Evaluation Labs and peer-reviewed Conferences. The Evaluation Labs are laboratories to conduct evaluation of information access systems and workshops to drive innovative evaluation activities. The conferences cover a broad range of issues such as continuation of Evaluation Labs activities, experiments using multilingual and multimodal data and research in evaluation methodologies and challenges.

3.1.2 CoNLL

SIGNLL [3], the ACL's Special Interest Group on Natural Language Learning, organizes CoNLL conference yearly focusing on theoretically, cognitively and scientifically motivated approaches to computational linguistics. CoNLL has been taking place since 1997 and during the years have covered many tasks in NLP domains including NER.

3.1.3 SemEval

The Special Interest Group on the Lexicon of ACL sponsors SemEval [8], a series of international NLP research workshop focuses on advancing the current state of the art in semantic analysis and creating high-quality annotated datasets in a range of increasingly challenging problems in natural language semantics. Each year, SemEval proposes several tasks to be completed and discuss in the conference.

3.1.4 IberLEF

Iberian Languages Evaluation Forum, IberLEF, [4] organizes competitive text processing, understanding and generation tasks to define new research challenges and setting new state of the art results for the NLP research community on any Iberian language: Spanish, Portuguese, Catalan, Basque or Galician.

3.1.5 BioNLP

SIGBIOMED, the Biomedical Natural Language Processing Special Interest Group of ACL, is dedicated to language processing in the biological, biomedical and clinical domain. The purpose of BioMed SIG is facing specific biomedical domain problems bringing together researchers in NLP, bioinformatics, medical informatics and computational biology.

One of the main promotion events is BioNLP [1] a workshop organized since 2004 that cover a wide range of NLP problems in biomedical domain.

3.1.6 ISCB

The International Society for Computational Biology [5] is an international non-profit organization that provides meetings, publications and reports on methods and tools of the

bioinformatics and computational biology domains. In conferences organized by ISCB, such as European Conference on Computational Biology, ECCB, tasks such as NER are covered.

3.1.7 LDC

The Linguistic Data Consortium, LDC, [6] is an open consortium of universities, libraries, corporations and government research laboratories founded in 1992. Since then, many resources and datasets have been published. One of the initiatives that drives LDC are conferences and workshops in different NLP techniques including NER.

3.1.8 LREC

The Language Resources and Evaluation Conference, LREC, [7] is an international conference organized by European Land Registry Association, ELRA, biennially. Since 1998, this conferences and workshops have been taking place and covering different NLP tasks.

3.2 Corpus and resources

In the following section, we will discuss some Corpus and NLP Systems available for the research community that could help as starting point for further work or as baselines to compare new NLP Systems and Corpus.

3.2.1 Corpus

There are multiple data sources for NER task available in the internet. Normally, these are domain focus, trying to be as specific as possible, in this section, only biological domain corpus will be discussed.

CORD-19

The Covid-19 Open Research Dataset, CORD-19, [88] is a growing resource made of scientific papers on Covid-19 and related historical coronavirus research with more than 377k entries. It includes metadata and structure full text papers to facilitate the development of text mining and information retrieval systems. It sources are PubMed Central, PubMed, the World Health Organization's Covid-19 Database and preprint servers bioRxiv, medRxiv and arXiv. Papers are processed in different steps, first they are clustered, then canonical

metadata is selected and finally not interesting entries are filtered. Also the PDF papers are parsed to be published as XML and JSON.

NCBI disease corpus

The NCBI disease corpus [31] presents a collection of 793 PubMed abstracts fully annotated at the mention an concept level of diseases to serve as a resource for the biomedical NLP community. Annotations were made manually following the Medical Subject Headings (MeSH) or Online Mendelian Inheritance in Man (OMIM) concepts. This corpus contains 6892 diseases mentions mapped to 790 disease concepts. Annotations follow PubTator guidelines.

CTD

The Comparative Toxicogenomics Database [27] was born in 2004 as a digital ecosystem that relates toxicological information for chemicals, genes, phenotypes, diseases and exposures to advance understanding about human health. This is information is literature-based and manually curated and now it contains 45 million toxicogenomic relationships for over 16300 chemicals, 51300 genes, 5500 phenotypes, 7200 diseases and 1630 exposure events from 600 comparative species.

PubMed Phrases

PubMed Phrases [46] is a collection of phrases beneficial for information retreival and human comprehension. There are 705915 PubMed Phrases extracted from PubMed publications. Statistical methods are applied to detect segments of consecutive terms that are likely to appear together in PubMed. Then, the quality of the data set is studied by analyzing a sample of 500 phrases.

NLM-Chem corpus

The NLM-Chem corpus [30] is a full-text resource to support the development and evaluation of automated chemical entity taggers. It consists on 150 full-text articles, doubly annotated by the expert NLM indexers. In total, it has around 5000 unique chemical name annotations mapped to around 2000 Medical Subject Headings (MeSH) identifiers.

MedMentions

MedMentions [66] is a manually annotated resource for the recognition of biomedical concepts. MedMentions uses 4000 annotated abstracts and over 350000 linked metions. Also, the concept ontology is based on Unified Medical Language System (UMLS) 2017 and covers over 3 million concepts.

SpanishCrawled

The Spanish Crawled corpus [20] or Corpus Web Salud Española (CoWeSe) is the largest Spanish biomedical corpus to date consisting of 750M tokens of clean plain text. It is the result of a massive crawler exercise on 3000 Spanish domains. The exercise started with 3338 manually curated links as seed with depth of 5, then only html headers and paragraphs were taken into account.

3.2.2 Natural Language Processing Systems

In this section, we present some Natural Language Processing systems able to solve the NER task. They have been developed by different researching groups present in the community and go from multi-platform APIs and interactive websites to open-source modules that can be used from programming languages like python. Also, in the last years interesting advances are taking place in the private sector. Different Cloud Computing providers such as Amazon Web Services, Microsoft Azure or Google Cloud are offering, between their products, cognitive services that usually include NLP systems able to accomplish NER task in many domains included biological one.

Metamap

MetaMap is a a highly configurable program developed at the National Library of Medicine (NLM) to map biomedical text to discover Metathesaurus concepts referred to in text. It uses knowledge-intensive approach based on NLP techniques. It can be used interactively, batch or API. Further information can be found in `https://lhncbc.nlm.nih.gov/ii/tools/MetaMap.html`.

Google Healthcare Natural Language API

Healthcare Natural Language API is part of the Google Cloud Healthcare API and it uses natural language models to extract healthcare information from medical text. Information that can be extracted includes medical concepts, such as medications, procedures, and medical conditions, functional features, such as subjects and relations such as side effects. It supports several medical vocabularies such as ICD-10, MedlinePlus Health Topics or Metathesaurus Names. Further information can be found in `https://cloud.google.com/healthcare-api/docs/concepts/nlp`.

Stanford Core NLP

CoreNLP [60] is a multitask system developed in Java. CoreNLP enables users to solve many NLP task such as NER among others. It supports 8 languages and is distributed under an GNU license. Its central piece is the Pipeline. This element gets raw text as input and produces a final set of annotations. Further information can be found in `https://stanfordnlp.github.io/CoreNLP/`

TwiCal

TwiCal [80] [81] is an open-domain event-extraction and categorization system for Twitter. This system is able to accurately extract an open-calendar of significant events from the social network. It is also able to discover important event categories and classify them. This system is developed in Python and is published under a GNU license. Further information can be found in `https://github.com/aritter/twitter_nlp`

NeuroNER

NeuroNER [28] is an engine based on a three layer LSTM Deep Learning architecture. The first layer is a character representation layer, the second a label prediction layer and, finally, a sequence optimization layer. It is developed in Python and distributed under a MIT license. Further information can be found in `http://neuroner.com/`

Polyglot

Polyglot is a python module that exposes a language pipeline that supports massive multilingual application such as NER in 40 languages. It is distributed under a GNU license, further information can be found in `https://polyglot.readthedocs.io/en/latest/index.html#`.

Neji

Neji is a biomedical information extraction platform that processes texts from scientific texts such as patents, publications and electronic health records. It can also create interactive web pages for NER tasks between others. It is developed in Java and distributed under a BDM Software license and University of Aveiro and a Creative Common license. Further information can be found in `https://github.com/BMDSoftware/neji`

NLTK

NLTK provides a flexible and simple framework for NLP tasks in python, this made it a NLP standard with python. It also provides a very simple toolkit for NER and relationship extraction. Further information can be found in `https://www.nltk.org/book/ch07.html`

spaCy

spaCy is one of the most popular NLP modules available in Python. It supports more than 66 languages and include many trained pipelines, transformers and word vectors. It has a specific component for NER task. Further information can be found in `https://spacy.io/api/entityrecognizer`

AllenNLP

AllenNLP [36] distributes a complete platform for solving NLP tasks such as NER in PyTorch. A broad collection of model implementations are offered and documented. This python module is continuously updated by the Allan Institute and distributed under a Apache license. Further information can be found in `https://allenai.org/allennlp/software/allennlp-library`

Chapter 4

Hypothesis statement and designed system

In this chapter, we will state the hypothesis that we will study in the experimental part of this work, as well as the description of the proposed system.

4.1 Hypothesis statement

Although NER systems performance have been improving a lot so far, and the number of NER tools have been increasing, we could see in historical review chapter (Chapter 2) that there is still room for significant improvement in NER task. Specially in complex domains, such as biomedical, and complex environment such as low resources environment. For this reason, many evaluation campaigns are organized. We will use one of those, the eHealth Knowledge Discovery Challenge at IberLEF 2021, as a framework to design our system and take advantage of its data set to study the training performance given sentences selection.

Golden standard annotated corpus generation is a very expensive task in terms of time and efforts. For this reason, usually it results hard to find annotated corpus for a certain NER task. However, for NER task systems performance to keep on improving, more labeled data is needed.

In this work we ask ourselves if it would be possible to prioritize those sentences that allow NER systems to learn quicker and better. If we were able to choose those sentences from a simple feature analysis criteria and annotate them before others that could not give so much information, we would be able to decrease annotation cost and get good NER task

results.

This idea comes in a moment when Deep Learning researchers and top voices are thinking about concepts like data labeling quality and data augmentation needs, in a movement called Data Centric AI (`https://datacentricai.org/`). This movement defends that good data is more important than much data when training a Deep Learning system. As well as, data augmentation is needed to deal with imperfections of real world data. Being able to have a prior knowledge of which sentences would have a better impact in system learning, would help to choose those that decrease data labeling and data augmentation costs, obtaining better results more efficiently because annotation and training time is reduced.

Thus, we state our hypothesis as follows.

Hypothesis

Some sentences in a corpus give more information to the NER system than others, and it is possible to distinguish the sentences that give more information from those that give less using simple linguistic features. Also, these sentences with more information lead to a better training efficiency.

We will study if previous hypothesis is language and length independent as well. Regarding the language independence condition, we can find universally accepted features that allow us to find the most interesting sentences for training in every language. This could have deep implications about how Deep Transformers abstract different languages grammar and vocabularies. About length sentence independent, we would observe similar tendencies when learning from short sentences or long sentences. This could be linked to how Deep Transformers apply attention techniques and how its context representations depends on sentence lengths.

4.2 System description

In this section, we will describe system proposed.

Collection	Source	Language	Size
Training	MedlinePlus	Spanish	1200
	Wikinews	Spanish	300
Develop	MedlinePlus	Spanish	25
	Wikinews	Spanish	25
	CORD	English	50
Testing	MedlinePlus	Spanish	75
	Wikinews	Spanish	75
	CORD	English	150
Total			1800

Table 4.1: Corpus sentences distribution [93].

4.2.1 NER in eHealth Knowledge Discovery Challenge at IberLEF 2021

We will base the experiment development on eHealth Knowledge Discovery Challenge at IberLEF 2021 data and methodology. eHealth Knowledge Discovery Challengee at IberLEF 2021 [93] is a NER campaign designed and focus on cross-domain, multi-lingual and low resource solutions. A corpus of 1800 English and Spanish sentences from biomedical and news domains are provided in order to face two NER tasks one of Entity recognition and another of Relation extraction. A number of 9 participants presented 10 different systems with varied levels of performance.

Corpus is token-level annotated with 4 entities, and 13 semantic relations and sentences have been extracted from Spanish MedlinePlus articles, Spanish Wikinews articles and English biomedical preprints related to COVID-19. Although Named Entities in this corpus are related to health topics, they show significant variety in terms of format and structure. Figure 4.1 shows how sentences are distributed in corpus. All data are available at `https://ehealthkd.github.io/2021/`.

We will focus on Task A which consists on identifying all the entities per document and their types. These entities can be single or multiple word and represent semantically important elements in a sentence. Entities will not present prefixes, suffixes or punctuation symbols. There are four types of entities:

- **Concept**: Identifies a relevant term, concept or idea in the domain of the sentence.
- **Action**: Identifies a process or modification of other entities, for example some verbs although nouns can also be actions.
- **Predicate**: Identifies a function or filter of another set of elements, which has a semantic label in the text and is applied to an entity with some additional arguments.
- **Reference**: Identifies a textual element that refers to an entity of the same or another sentence.

As it can be seen, these entities do not refer to specific biomedical concepts such as certain diseases. For example, there is not an entity for flu, another for covid and others for other diseases. Instead of this, these entities refer to more general information that could help to find relationships in texts. This also makes this data set more interesting because the methodology to solve this task could be directly exported to find entities with this general information to find relationships in other domain corpus.

We focus on this campaign methodology because it gives us an standard to develop a biomedical and news domain system in an multilingual and low resource environment which is itself a state-of-art task. Also, it would allow us to compare the performance of our solution to other proposed systems in a controlled environment. This campaign also gives us a well annotated data set in a low resource environment which is fundamental to give a good context for the study of the impact of sentence selection in NER task system training.

For task A, we have designed a system based with three steps, (1) a pre-processing one that annotates the Corpus given the annotation files using BIO labels, (2) the training step in which a BERT transformer learns from corpus how to classify the different classes and (3) the post processing phase that corrects some possible defects of BIO tagging framework. In the further sections, we will discuss in more depth the corpus pre and post-processing and model definition steps. In Chapter 5, we will describe training and evaluation steps.

4.2.2 Corpus pre-processing

Annotation files are pre-processed to assign a BIO label to every entity prior to beginning the pre-processing step. This is straightforward given that every entity is labeled with its

entity label, so first word in the entity gets the B label and, the rest, the I label. This BIO labels are complemented with the entity type. For example, if we have the first word of a concept entity, we will tag it as B-concept. Although in the annotation files there are no words with an O tag, every word in corpus that does not appear as an entity or part of an entity in annotation corpus will be tagged as O in the pre-processing step.

Once we have pre-processed the annotation files, we start the corpus pre-processing step. We first detect which sentences are English and which ones are Spanish, we do this using the *langdetect* python module.

Then, we process the sentences in order to get their POS tag using the Spacy module and the pre-trained models es_core_news_sm and en_core_news_sm. Once we have the sentences tagged we start the annotation process.

The annotation process iterates over every word in the corpus and checks if it is annotated in the following 20 words of the annotation files. In case that the corpus word has a match in the annotation files, we use its label, otherwise, we jump to the next corpus word. Doing this, we can process the whole corpus and simple entities will not pose a problem, i. e. one group of words only leads to one entity. However, if there are some group of words in the corpus that lead to more than one entity, BIO schema could fail. For example, the corpus sentence: *El dolor puede comenzar uno o dos días antes de su período* contains the words *uno o dos días* which leads in annotation files to two entities: *uno días* and *dos días*. If we had processed the text just in a sequential way, as explained before, we would have got stuck and BIO tagger pre-processing would have been misleading. In Figure 4.1 this problem can be observed. Given this example, it is easier to understand why the system looks for the target word in the corpus between the following 20 words in annotation files. Also, if one annotated entity has not been assigned to any corpus word during 100 iterations, we delete it from the 20 annotated entities window in order to free space for other entities. This algorithm allow us to process the whole corpus correctly.

At the end of this step, each word in corpus has its POS tag and BIO label. This is a general technique that we apply to those data sets that we are using for training and development. The exact data that we are using depends on the experiment that is taking place. We will give more detail about the methodology in chapter 5.

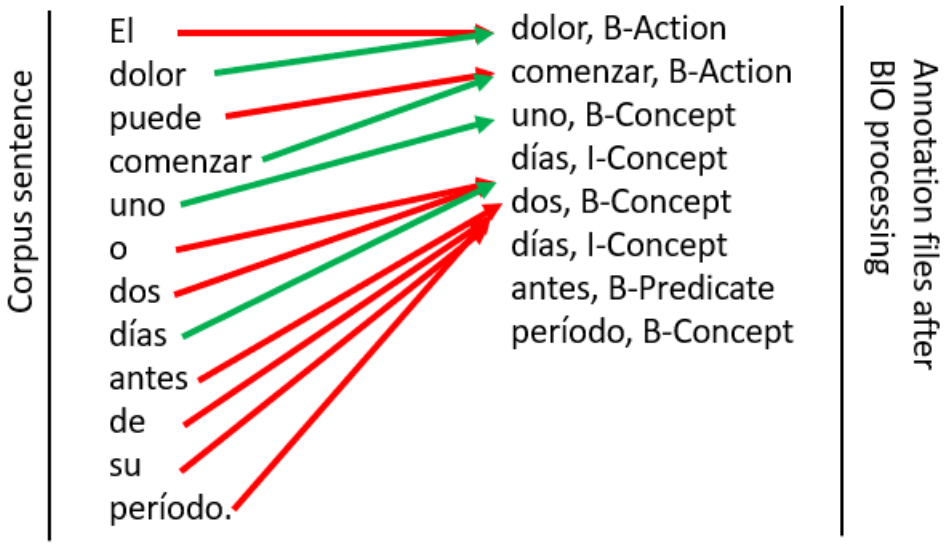

Figure 4.1: Sequential BIO Tagger problems when facing multiple entities in the same group of words.

4.2.3 Model definition

Once we have the data set tagged and labeled, we can define our system model. We have used a BERT model pre-trained with *bert-base-uncased* [29] from Pytorch module. One model for English sentences and another one for Spanish are trained. Also, we define two systems depending on the training sentences length used as follows:

All lengths in one model

For a simple configuration, we will train one model for all length sentences.

Two lengths in two models

As seen from corpus analysis section (Section 5.2.1), we find two length domains in corpus, one from 0 to 25 words and another one from 25 words in advance. Although BERT is trained using attention mask, first experiments trying to train the system showed that performance raised when these two domains are modeled separately, it can also be seen in the final results. We will use this configuration to study length dependence in sentence selection. This gives us four BERT models for each prediction that get in use depending on the length and language of the sentence.

Although there are other pre-trained BERT models, such as multilingual bert M-BERT,

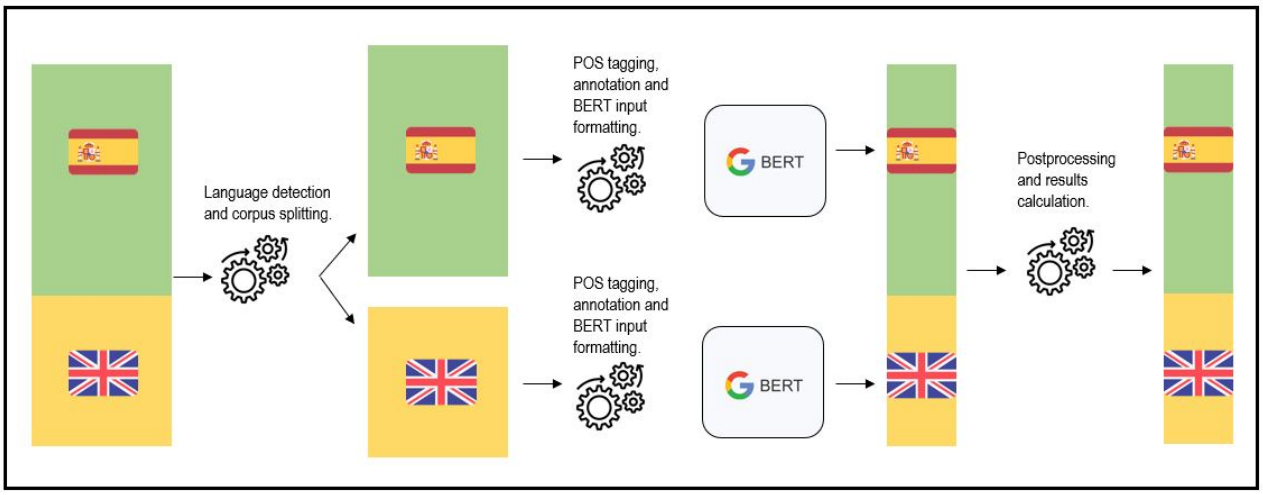

Figure 4.2: Architecture schema with data flows and processing components.

that could have had better performance, given that they have been trained over dozens of languages, we have used the only english version in order to try to aisle possible multilingual effects in our study of sentence selection impact in training.

4.2.4 Post-processing

After model training and evaluation, we need to post-process the results in order to correct some possible defects in prediction given the BIO label schema. This consists on making two actions:

- If the predicted label for one word is an Inside label, but previous label is not a Beginning label, then we replace, the I label for a B label.
- If the predicted label for one word is an Inside label and the previous label is a Beginning label, but both don't share entity type, then we replace I label entity for the one in the B label.
- If the predicted label for one word is an Inside label and the previous label is an Inside label, but both don't share entity type, then we replace the second I label entity for the one in the first place.

Figure 4.2 depicts a high-level schema of the system architecture.

Chapter 5

Experiment description and results

In this section, we will describe the experiment configuration and the obtained results.

5.1 Technical Environment

All steps in system and experiment creation and development have been deployed under a Google Colab PRO license with Python 3.7 language and GPU back end. Pytorch and Spacy have been used for the model creation and training and language processing steps.

All code is available at `https://github.com/eledeluisest/TFM_SentencesNER`. The ordered execution of all codes would lead to the analysis and results that can be seen in this work. As more detailed information, every run of the experiment takes around 8 hours, so parallel execution is advised.

5.2 Experiment description

Two main experiments have been conducted. For the first one, we try to prove the hypothesis following these steps.

First, as this system won't be use for a eHealth-KD submission comparative, all data available, i.e. training, development and testing corpus, are analyzed so interesting information for feature extraction and system definition is found.

Second, we create our own fixed development data set out from all data available as the 20% of each available corpus (CORD, Medline and Wikinews). This will allow us to measure the impact of sentence selection.

Finally, we train and evaluate the system over several different situations depending on the sentences chosen to study how this could drive to a better or worse performance, these sentences are selected from data that was not chosen for development data set. We will discuss selection criteria in more detail in the following section.

The second experiment consists in an eHealth Knowledge Discovery Challengee at IberLEF 2021 submission. Using the system designed for the previous experiment, we will train two systems, one using all data set available in training and development corpus, avoiding using test data set, and another one using only certain selected sentences in training and development corpus, avoiding using test data set as well.

Later, we will check their performance on the test corpus, which has not been used for training, so we can have an approximate rank position in the evaluation campaign for our system when using all available sentences or only certain ones. This experiment does not have an explicit section because following the commented methodology and knowing the architecture in use the application is direct.

5.2.1 Corpus Analysis and Linguistic Features Extraction

We have extracted three characteristic features for each sentence using well known techniques such as computing word frequencies in corpus, computing POS frequencies in corpus and applying a semantic embedding representation to get sentence meaning information.

We looked for some straightforward criteria to distinguish the sentences in terms of lexical, morpho-syntactical and semantic information. During the data observation, we could check that there were sentences with few words very infrequent and other sentences with many words not that infrequent, that led to similar measures of mean word or POS frequency. One example of this can be seen in Table 5.1. Both sentences are included in Medline 1200 corpus and both have similar mean word frequency. Sentence number 1189 has a $1.39 \cdot 10^{-2}$ mean word frequency and sentence number 423 has a $1.29 \cdot 10^{-2}$ mean word frequency.

We are trying to find those sentences that would provide first the most specific and less frequent information to the system. For this reason, we designed a metric that allowed us to distinguish the two situations that we explained before in favor of that sentences that have few words very infrequent. In the example above, we would like to prioritize sentence number 1189 over sentence number 423, because, as it is shown by columns *Word appearance*

Sentence #	Word	POS	Tag	Word appearance frequency
1189	los	DET	O	2,56E-02
1189	procedimientos	NOUN	B-Concept	2,53E-04
1189	de	ADP	O	5,58E-02
1189	tecnología	NOUN	B-Concept	2,53E-04
1189	de	ADP	O	5,58E-02
1189	reproducción	NOUN	B-Concept	1,27E-04
1189	asistida	ADJ	B-Concept	1,90E-04
1189	a	ADP	O	1,35E-02
1189	veces	NOUN	O	8,86E-04
1189	usan	VERB	B-Action	4,43E-04
1189	óvulos	NOUN	B-Concept	6,33E-05
1189	de	ADP	O	5,58E-02
1189	donantes	NOUN	B-Concept	6,33E-05
1189	un	DET	O	1,33E-02
1189	donante	NOUN	B-Concept	1,27E-04
1189	de	ADP	O	5,58E-02
1189	esperma	NOUN	B-Concept	1,27E-04
1189	o	CCONJ	O	1,40E-02
1189	embriones	NOUN	B-Concept	6,33E-05
1189	previamente	ADV	O	6,33E-05
1189	congelados	ADJ	B-Concept	6,33E-05

Sentence #	Word	POS	Tag	Word appearance frequency
423	el	DET	O	3,27E-02
423	tratamiento	NOUN	B-Action	3,23E-03
423	para	ADP	O	9,18E-03
423	algunos	DET	O	1,77E-03
423	tipos	NOUN	B-Predicate	1,08E-03
423	de	ADP	I-Concept	5,58E-02
423	cáncer	NOUN	B-Concept	2,72E-03
423	de	ADP	O	5,58E-02
423	garganta	NOUN	I-Concept	6,33E-04
423	también	ADV	O	3,67E-03
423	puede	AUX	O	1,12E-02
423	incluir	VERB	O	1,14E-03
423	terapia	NOUN	B-Concept	9,49E-04
423	dirigida	ADJ	B-Concept	2,53E-04

Table 5.1: Example of representative sentences in Medline 1200 corpus.

frequency, has more number of infrequent words that could help our system to learn better the corpus particularities.

Now, we need to define what *very infrequent* means. We studied the word and POS frequency in corpus for every corpus. As a representative example, in Figure 5.1, we show the analysis of Medline 1200 corpus for word frequency. As it can be seen in the whole data range histogram, there are few words that have very high word frequency values. Those are stop words and other very common words inside the data set.

In order to only take into account the very infrequent words, we study how the distribution behaves in the lower frequency values. In the right histogram in Figure 5.1, only values from 0 to 0.0001 are represented. The vertical lines are the distribution percentile values. We chose as very infrequent those with a lower to percentile 33 frequency, because it offers a good balance between frequency values and number of words selected as *very infrequent*. In the one hand, selecting percentile 10 wouldn't have selected almost any word. In the other hand, choosing percentile 33 to percentile 25 enables us to select more words without gaining to much frequency value.

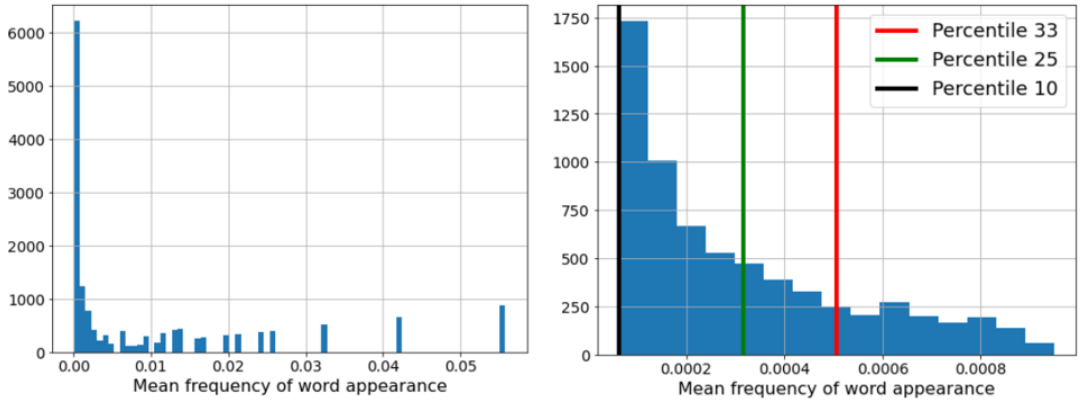

Figure 5.1: Word frequency distribution in Medline 1200 corpus. We show the distribution in two different scales, one for the whole data range and another one in the lower frequencies range.

In Table 5.1, for the two representative sentences, we give a green color to those *very infrequent* words. It can be seen, how this definition helps us to choose the sentences with most infrequent words.

In the case of semantic features, we follow the same logic and try to distinguish those sentences that are further in meaning from the mean meaning of the corpus. In this case, the vectorial space gives us tools to measure differences between vectors, we have chosen the module difference. In more detail, the three features that have been extracted are the following.

- **Word frequency in corpus.** For each word in corpus, the relative frequency of appearance is computed. Then we compute percentile 33 in word relative frequency distribution and count, for each sentence, the number of words that appear less than percentile 33. With this count we create the first feature: sum_freq_palabra.

- **POS frequency in corpus.** For each word in corpus, the relative frequency of its POS tag appearance is computed. Then we compute percentile 33 in POS tag relative frequency distribution and count, for each sentence, the number of POS tags that appear less than percentile 33. With this count we create the second feature: sum_freq_pos.

- **Sentence semantic embedding.** For each word in corpus we compute its embedding using Word2Vec and *word2vec-google-news-300* model. Then we compute the

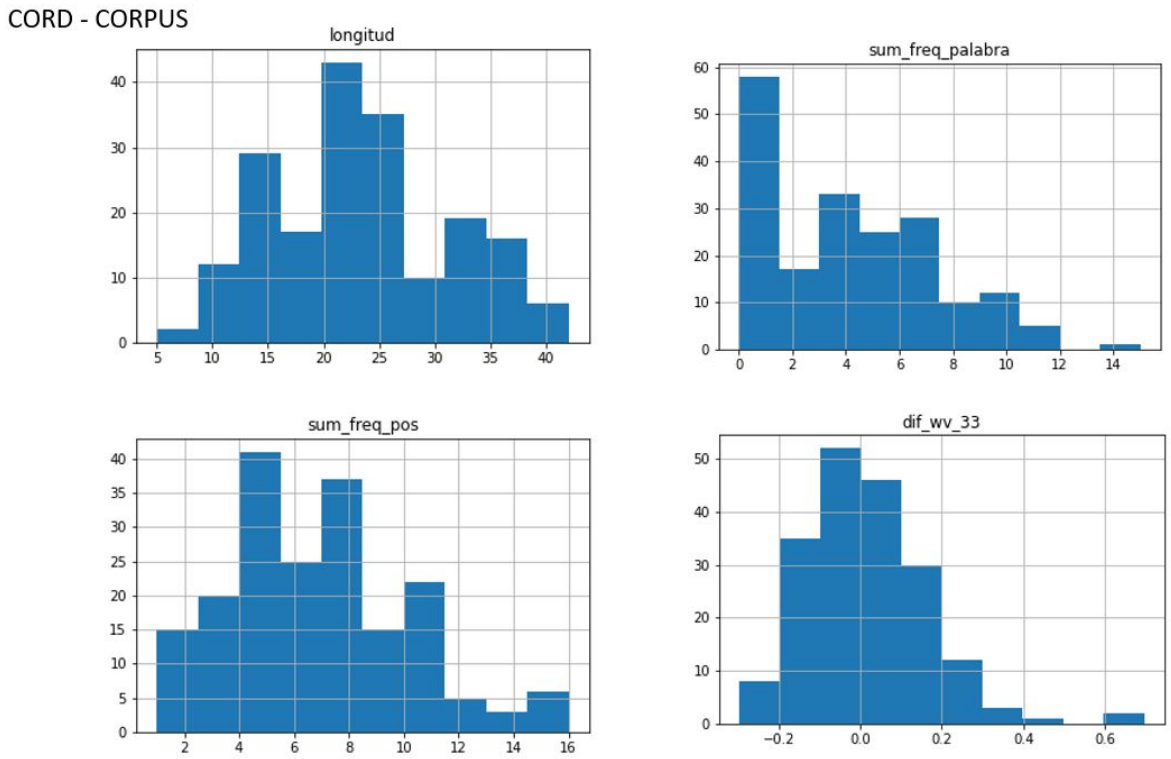

Figure 5.2: Cord corpus distribution of sentence features.

module of the mean embedding for each sentence. After this, we compute the difference between this mean module and the median of the sentences mean embedding distribution. With this difference we have the third feature: dif_wv_33

In the following images, we show the distribution of this three features as well as the sentence length for each corpus in the data set, CORD (Figure 5.2), Medline (Figure 5.3) and WikiNews (Figure 5.4).

As it can be seen, the largest sentence length for Spanish corpus (medline and wikinews) is 25 for medline and 50 for wikinews while English corpus (cord) max length is 24. Now, if we pay attention to features distribution we see that it has similar shapes in Spanish corpus and a little different in English corpus. sum_freq_palabra distribution has more left-weight in the English corpus, sum_freq_pos distribution has less left-weight in the English corpus and dif_wv_33 is almost symmetric in Spanish corpus but left-weighted in English corpus. Also, maximum values for sum_freq_pos and sum_freq_palabra is more than half of length in CORD and Medline corpus but less than half of length in WikiNews, the reason could be that WikiNews vocabulary is less diverse than those of Medline and CORD.

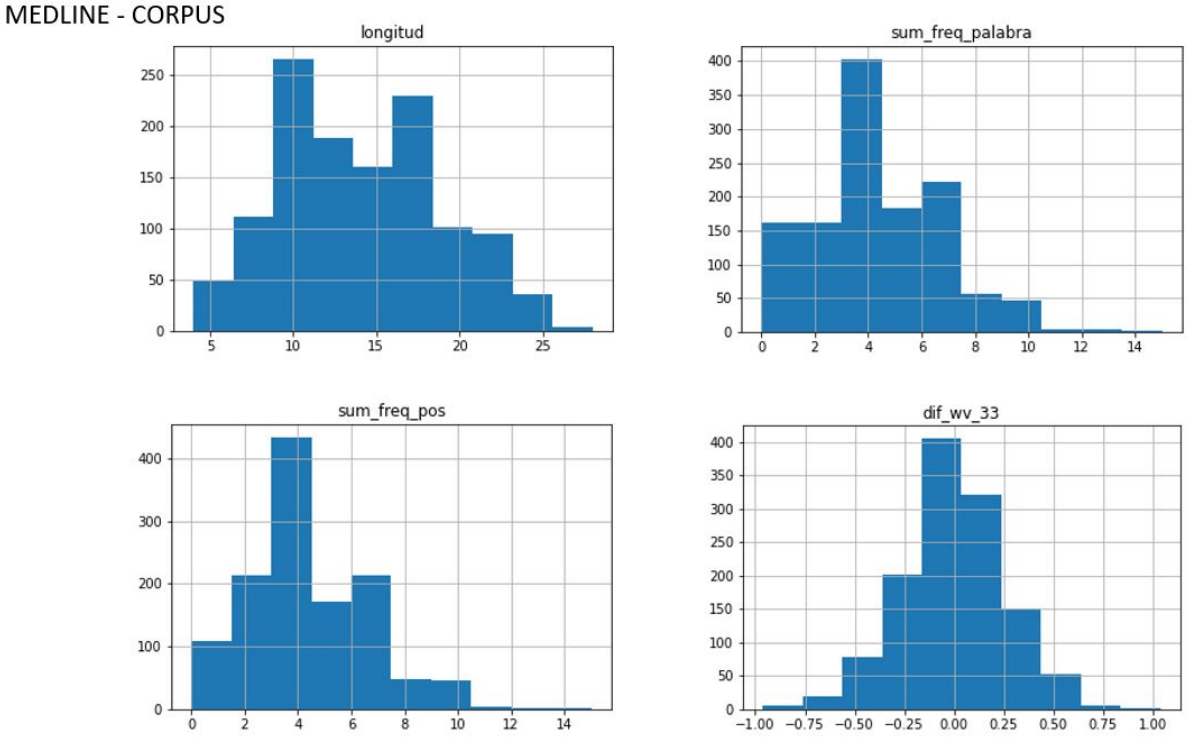

Figure 5.3: Medline corpus distribution of sentence features.

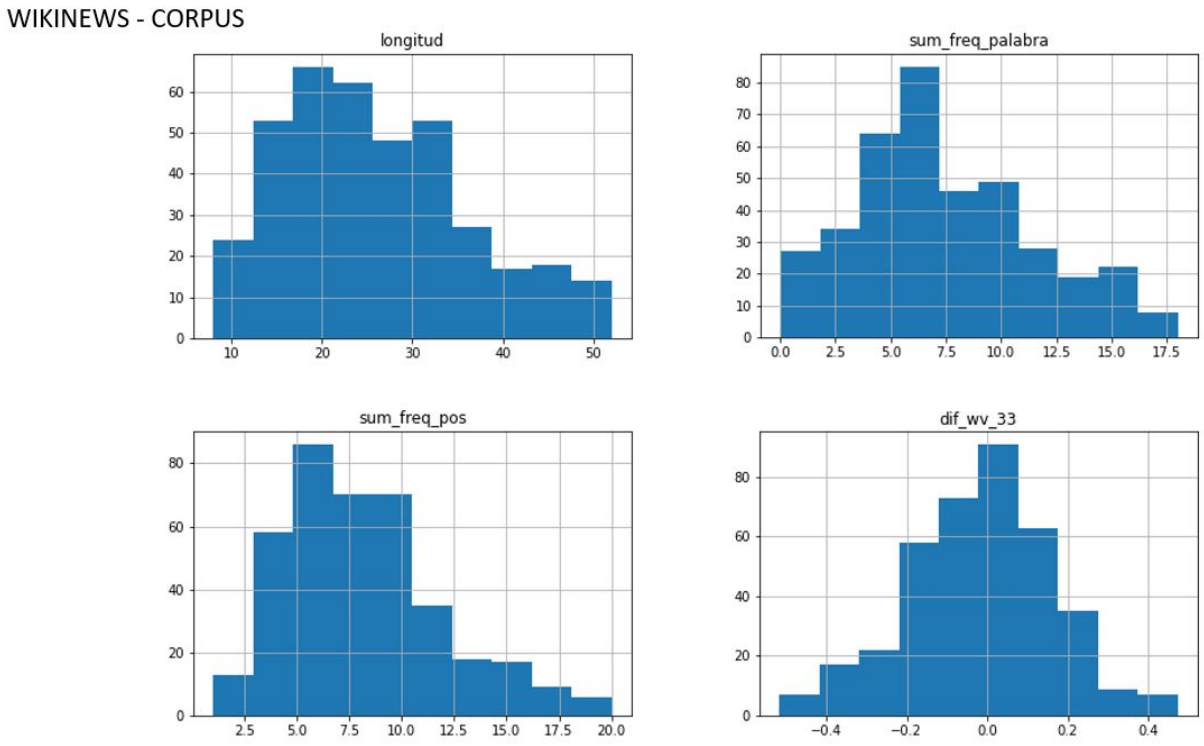

Figure 5.4: Wikinews corpus distribution of sentence features.

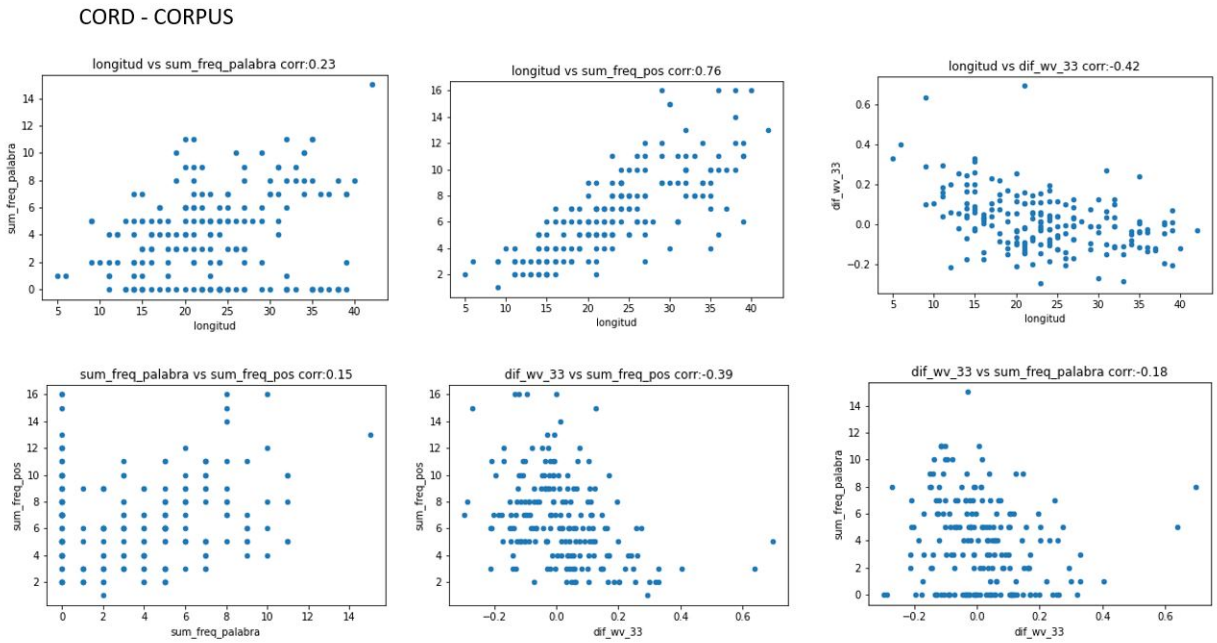

Figure 5.5: Cord corpus sentence features correlations.

Now, for each corpus we will compute correlation coefficient for this 4 features one against one. We would like to check that does not exist strong dependencies between length and the other features so they are length independent. We want also make sure that does not exist strong correlation between features, so choosing top sentences for every feature would lead to different sentences.

In Figures 5.5 for CORD, 5.6 for MedLine and 5.7 for WikiNews, results from correlation analysis can be seen. Some correlation can be observed between length and sum_freq_pos and sum_freq_palabra. In CORD corpus this correlation is moderate (0.76) between length and sum_freq_pos, in MedLine corpus this correlation is moderate (0.6) between length and sum_freq_pos and in WikiNews corpus this correlation is moderate between length and sum_freq_pos (0.68) and between length and sum_freq_palabra (0.71). This correlations are not strong enough (less than 0.8) to demonstrate that these features are length dependent or that choosing top sentences for each feature would lead to the same sentences.

5.2.2 Training and evaluation methodology

We will describe training parameters and methodology jointly for all sentences in one model configuration and for two length sentences in two models. Both share parameters and the

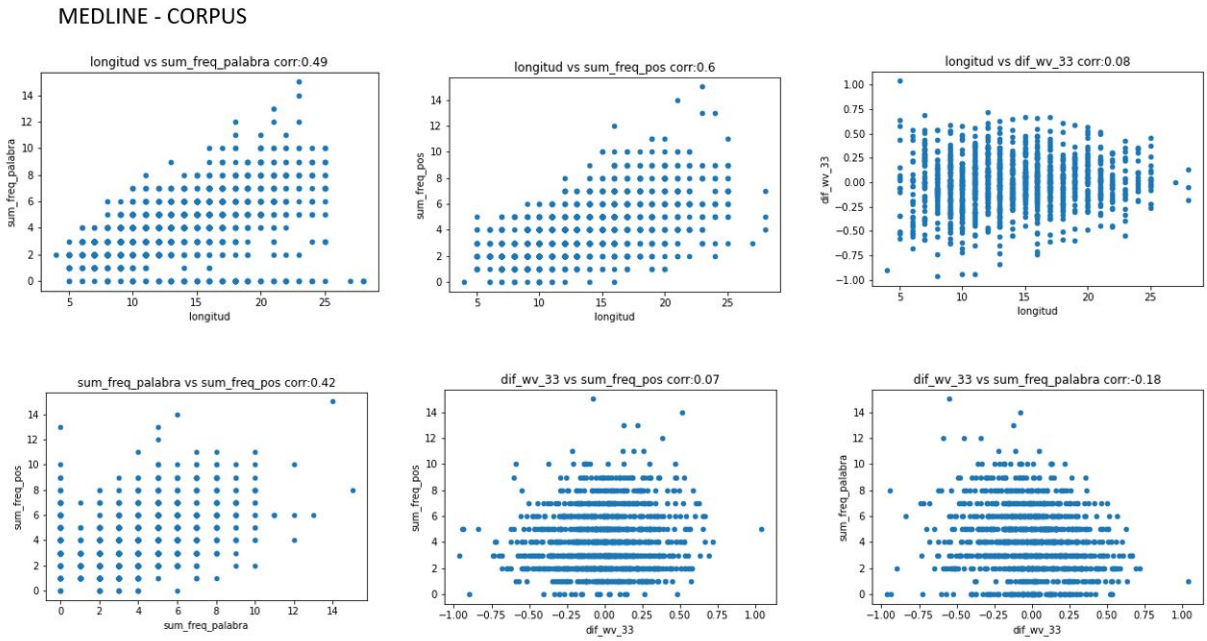

Figure 5.6: Medline corpus sentence features correlations.

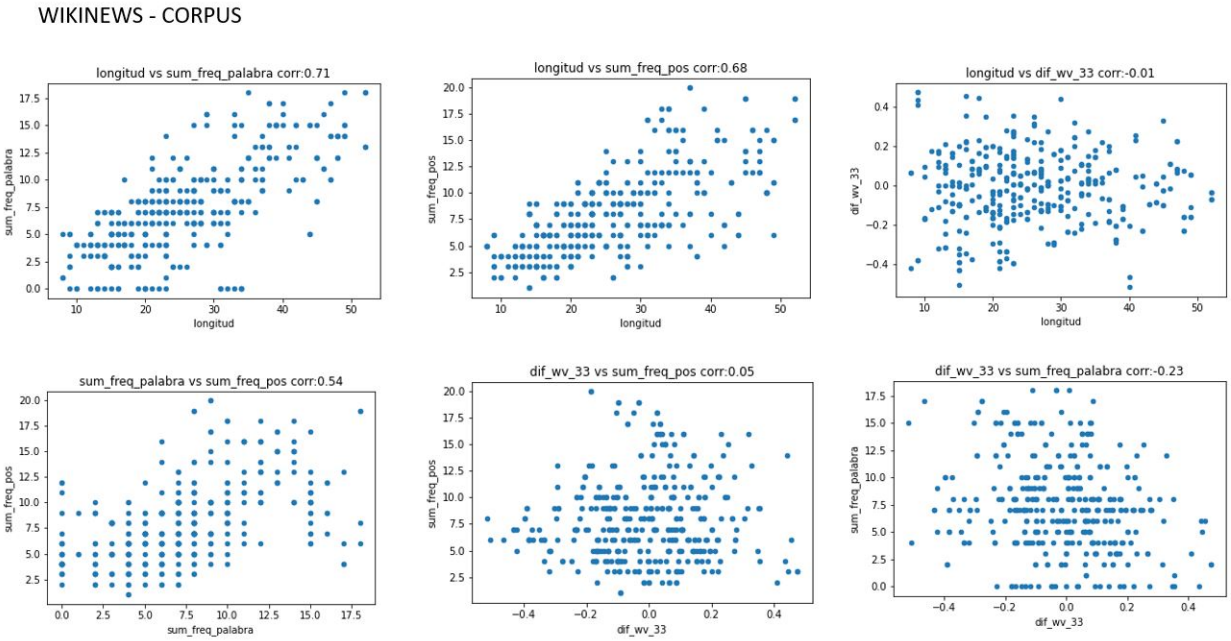

Figure 5.7: Wikinews corpus sentence features correlations.

only difference is one split between lengths domains before training phase one for sentences shorter than 25 words and another one for sentences between 25 and 50 words.

First, we create one fixed development data set made from a 20% random selection of competition training, development and test corpus. As we want to study the performance of the system depending on the number of sentences extracted given some certain criteria that we use to train, we create data sets with top 150, 300, 450, 600, 750, 900 and 1200 sentences given each feature and one random selection. This is 150, 300, 450, ... sentences with smaller sum_freq_palabra value 150, 300, 450, ... sentences with smaller sum_freq_pos, 150, 300, 450, ... sentence with biggest absolute diff_wv_33 and 150, 300, 450, ... random sentences.

Then, all sentences go through a formatting phase to be represented in BERT input format, which is a numeric array. This numeric array is designed from an indexed vocabulary made from training data set, out-of-vocabulary words will be represented as value 1 and label *UNK* and array positions that don't have associated word have value 0 and label *PAD*. This padded position are combined with an attention mask that ensures that BERT only focuses in those positions that have a word informed. Then, with the arrays and its related labels pytorch dataloader classes are created. Once we have this classes created, the training phase can start.

These are the main parameters for training and BERT configuration:

- Full fine tuning of bert-base-uncased pre-trained BERT with a weight decay rate of 0.01.
- We use Adam optimizer with a learning rate of $3 \cdot 10^{-5}$ and an scheduler that helps reducing learning rate a 2% over the last learning rate value each epoch.
- Loss function is calculated from the standard *compute_loss* method from pythorch model class and back propagated.
- We train with 450 epochs and a batch size of 16 sentences.

Once one model is finished with the training phase, training and development data sets are evaluated. The results, a relation of all predicted and observed labels, are recorded for a further analysis as well as the model for backup and reproducibility. Predicted labels are defined as the argmax of the output array of BERT which gives a probability for each

possible label (B-Reference, I-Reference, B-Concept, I-Concept, B-Predicate, I-Predicate, B-Action, I-Action, O).

After evaluation, post-processing phase takes place. Then, calculations are made to align final results with eHealth Knowledge Discovery Challenge campaign evaluation metrics. These are the different result metrics [93]:

- **Correct, C**. We get a correct result when predicted and target label completely match.
- **Partial, P**. We get a partial result when predicted and target label partially match.
- **Incorrect, I**. We get an incorrect result when predicted and target label are both entities (i.e they are not O) but they don't match.
- **Missing, M**. We get a missing result when predicted label is O but target label is an entity.
- **Spurious, S**. We get a spurious result when predicted label is not O but target label is not an entity (i.e it is O).

With these calculations, we can compute the precision, recall and f1 metrics, which are the eHealth 2021 metrics [93], following the following equations:

$$precision = \frac{C + 0.5 * P}{C + P + I + S} \tag{5.1}$$

$$recall = \frac{C + 0.5 * P}{C + P + I + M} \tag{5.2}$$

$$f1 = 2 * \frac{precision * recall}{precision + recall} \tag{5.3}$$

We iterate this process 5 times to have enough data to do some statistics and have variation information for each point.

5.3 Results

In this section, we will present the results of the experiment. We will plot f1 score, precision and recall values for each number of sentences studied: 150, 300, 450, 600, 750, 900 and

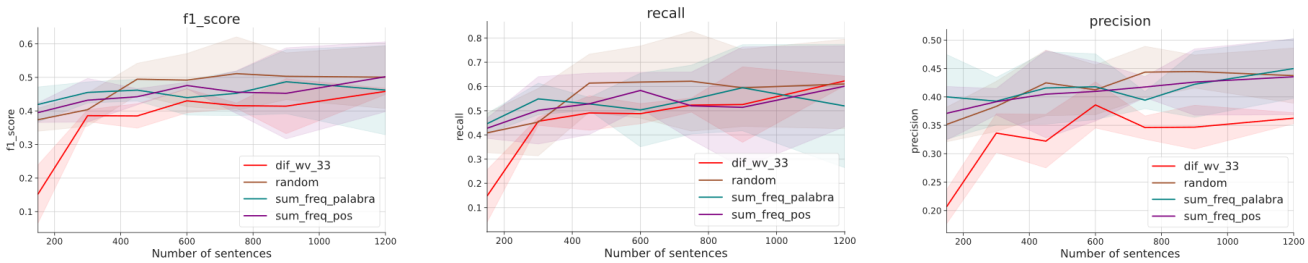

Figure 5.8: Results on validation data set for one length system. Average and standard deviation results over five runs are represented under filled lines and shadow areas respectively.

1200. Also we will draw one line for each sentence selection criteria, which are: biggest sum_freq_palabra, sum_feq_pos, largest absolute diff_wv_33 and random. Average and standard deviation results over five runs are represented under filled lines and shadow areas respectively.

In Figure 5.8, we see the results of the trained system with just one length domain. As it can be observed, for small number of sentences, below 450, explicit sentence selection perform better than random selection but, for larger number of sentences, random selection is able to perform better in every metric except precision in which sum_freq_palabra, sum_feq_pos and random are relatively similar until 750 sentence in training data set.

Results for one length system and English and Spanish data are represented in Figure 5.9. For Spanish data, we see similar tendencies as for complete data set whereas for English data, we observe that random selections perform better in an even stronger way for points over 150 sentences.

Let's see now the results on validation data set for two lengths system. In Figure 5.10, we see general results for two lengths system, here we also see how for small number of sentence below 450, sum_freq_palabra and sum_freq_pos perform better than random selection.

If we analyze the impact of languages in the two length system (Figure 5.11), we see that again training in Spanish data gives similar results to those in general configuration. For English, we see that sum_freq_pos is able to perform better than other criteria including random for almost every point.

Lastly, we study the impact of length in the results in Figure 5.12. For the short domain, we see very similar results to general ones. In contrast, for long domain we see how

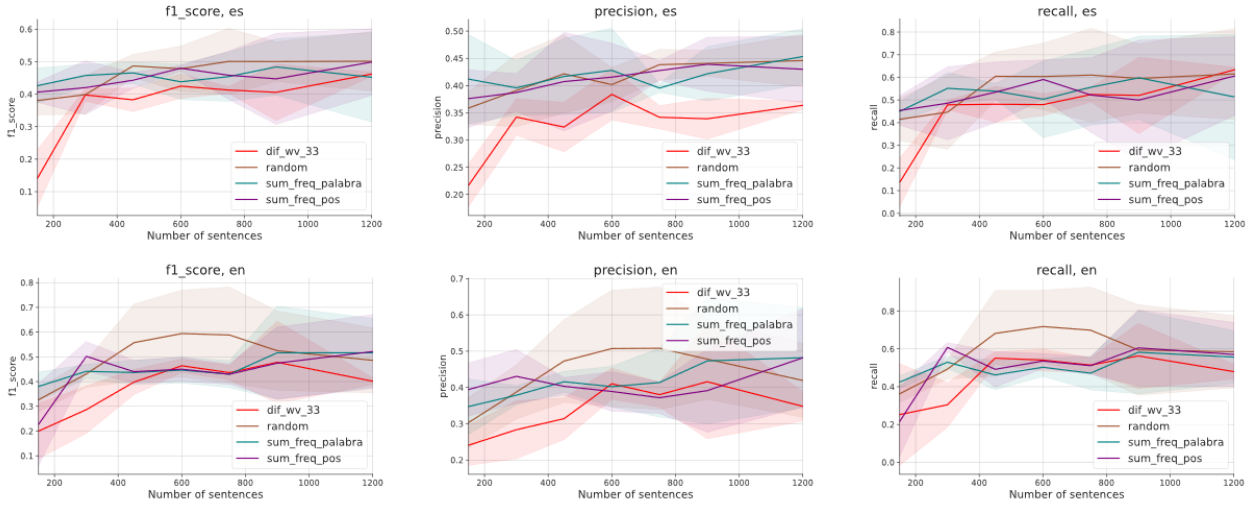

Figure 5.9: Results for English and Spanish data on validation data set for one length system. Average and standard deviation results over five runs are represented under filled lines and shadow areas respectively.

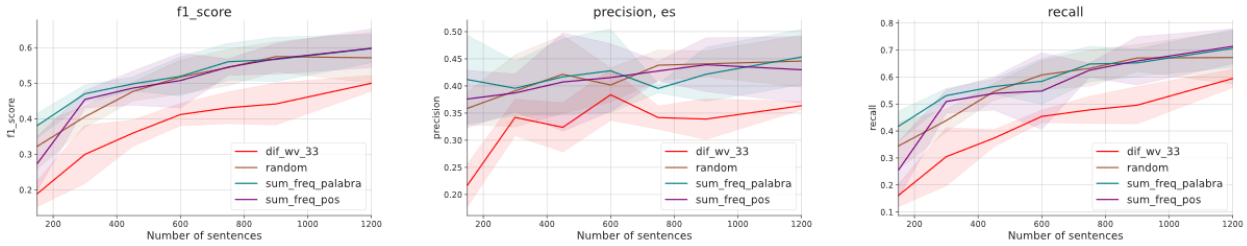

Figure 5.10: Results on validation data set for two lengths system. Average and standard deviation results over five runs are represented under filled lines and shadow areas respectively.

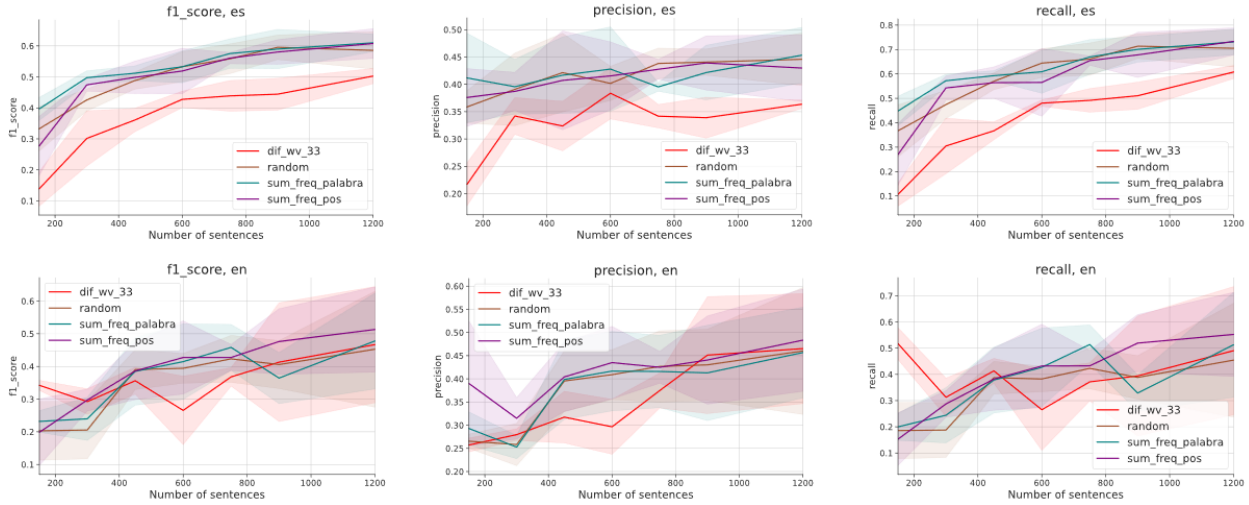

Figure 5.11: Results for English and Spanish data on validation data set for two lengths system. Average and standard deviation results over five runs are represented under filled lines and shadow areas respectively.

sum_freq_palabra and sum_freq_pos perform much better than random selection for small number of sentences, below 450.

5.4 Discussion

First, regarding the hypothesis, it seems that for small corpus it is possible to choose those sentences that would make the system learn faster. This could be interpreted as a help to fine-tuning process to make the system focus only on most significant features.Anyways, as we increase the data set size, random selection performs equal or even better than other selection criteria. This can be explained because bigger random selection include more sentences from other selections with other criteria.

Also, we can clearly see in the results how morphological and syntactical features are more useful than semantic embedding ones. However, it is interesting that choosing the sentences with largest differences in semantic representation always performs worse than random selection. This could suggest that selecting sentences with the opposite criteria could lead to better results.

Second, we saw in results that sentence selection is dependent on language and length. In the first case, we can see different behavior for English and for Spanish. Also we see

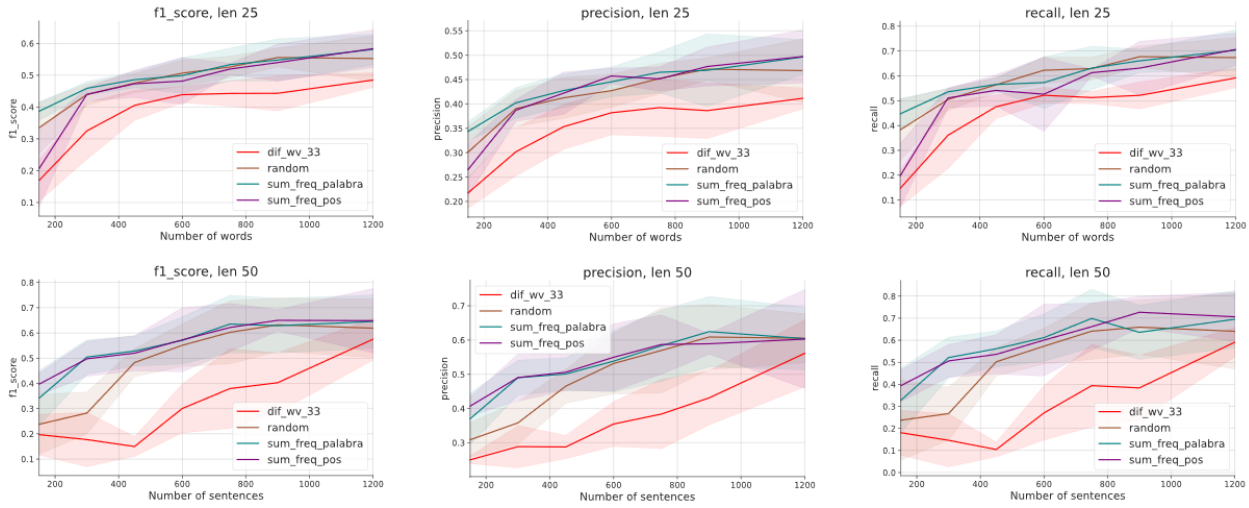

Figure 5.12: Results for short and long domain on validation data set for two lengths system. Average and standard deviation results over five runs are represented under filled lines and shadow areas respectively.

that logically general results are very similar to Spanish ones because this language is more frequent in the data set. Anyways, for small corpus we still see that sum_freq_pos criteria can perform better than random selection in English.

Regarding length dependence, we see how for short sentences random and non-random selection criteria lead to similar results, but for long sentences we can see how sum_freq_pos and sum_freq_palabra perform better until the biggest corpus sizes. One reason for this could be that in this case length discriminates effectively the corpus, as can be seen in corpus distribution analysis, longest sentences belong to WikiNews corpus. This would reinforce the idea, that choosing certain sentences for training can help system to learn particular corpus characteristics. This could be because, BERT model in use had been pre-trained with general domain corpus like WikiNews, so this would suggest that sentence selection is more effective when system is pre-trained in the same domain of sentences to be selected. This follows the common dynamics of fine tuning.

It is also important to notice, that two length system performs generally better than one length system. Given that both have the same attention methods, it seems that allowing too much padding decreases attention performance. In one length system, most sentences have more padding than actual words as can be seen in length distribution analysis.

Finally, we give the results of the second experiment, which consisted on evaluating our

system following the evaluation campaign criteria. After re-training the system used for the other experiment completely, avoiding the use of testing data, we have compared our best performing system setting which is two lengths system with 1200 sentences and sum_freq_pos criteria. It scored 0.192 for f1 score, 0.181 for precision and 0.205 for recall. These results in test data set would lead our system to a 7th (depending on sorting criteria) place out of 9 participants on eHealth Knowledge Discovery Challenge Task A. However, results turn out to be worse than baseline. Also, we have re-trained our system with all training and development data available, avoiding the use of testing data. We obtained the following results on test: 0.228 for f1 score, 0.205 for precision and 0.257 for recall. As it can be seen, difference between whole data set training and the reduced one is only of 0.036 (15%), while difference in number of sentences is about 50% and also training times are almost double. Although performance difference is remarkable, it is not big enough to invalidate that in low resources environments it could be worthy choosing this sentences before. Although the system presented by competition winners obtained much better results, f1=0.706 for task A winner team: PUCRJ-PUCPR-UFMG [93], these systems are also based on Deep Transformer BERT with different pre-trained configurations.

Chapter 6

Conclusions and further research

As has been proved, using Deep Transformers allows good performance for complex NLP tasks such as NER. However, a considerable investment has to be done in pre-processing and post-processing. This could open a research line to define a unique annotation standard. This standard would attempt to simplify pre and post processing phases. Furthermore, making it hierarchy complex enough would enable us to apply it to many different problems, hence enabling comparisons across many tasks.

Also, Deep Transformers need a heavy infrastructure in the form of GPUs and TPUs to train fast enough. This creates a *learning gap* between research teams, companies and societies that have access to these kind of technologies and those who don't. This, in the mid-term could make bigger gaps between economies and societies. Paradoxically, data applications that could close gaps between developed and not developed societies could make it even bigger.

About the hypothesis, the obtained results don't deny it but don't validate it clearly. Further research in small number of sentence would be required. Also, defining other features, specially semantic features, but also morphological and syntactical, could lead to a deep understanding of sentence selection impact in training phase. I would also be interesting to study this hypothesis using other BERT configurations, such as multi-lingual BERT or other deep transformers like GPT-3.

Finally, this kind of work could be understood inside the Data Centric AI movement which was officially founded in the year that this work is presented, 2022, and has a promising future ahead.

We would like to use this last lines of the conclusion section to kindly apply for an extra point for this work, given it has been written in English, with a quality that the author finds acceptable.

Bibliography

[1] Bionlp. https://aclweb.org/aclwiki/BioNLP_Workshop.

[2] Conference and labs of the evaluation forum. http://clef-initiative.eu/.

[3] Conference on natural language learning. https://conll.org/.

[4] Iberlef. https://sites.google.com/view/iberlef2020/.

[5] Iscb. https://www.iscb.org/index.php.

[6] Ldc. https://www.ldc.upenn.edu/about.

[7] Lrec. http://www.lrec-conf.org/#.

[8] Semeval. https://semeval.github.io/.

[9] Stein Aerts, Diether Lambrechts, Sunit Maity, Peter Loo, Bert Coessens, Frederik De Smet, Léon-Charles Tranchevent, Bart De Moor, Peter Marynen, Bassem Hassan, Peter Carmeliet, and Yves Moreau. Gene priotization through genomic data fusion. *Nature biotechnology*, 24:537–44, 06 2006.

[10] Gustavo Aguilar, Suraj Maharjan, Adrian Pastor López-Monroy, and Thamar Solorio. A multi-task approach for named entity recognition in social media data. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pages 148–153, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

[11] Alan Akbik, Duncan Blythe, and Roland Vollgraf. Contextual string embeddings for sequence labeling. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1638–1649, Santa Fe, New Mexico, USA, August 2018. Association for Computational Linguistics.

[12] Enrique Alfonseca and Suresh Manandhar. An unsupervised method for general named entity recognition and automated concept discovery. In *In: Proceedings of the 1 st International Conference on General WordNet*, 2002.

[13] Bogdan Babych and Anthony Hartley. Improving machine translation quality with automatic named entity recognition. In *Proceedings of the 7th International EAMT workshop on MT and other language technology tools, Improving MT through other language technology tools, Resource and tools for building MT at EACL 2003*, 2003.

[14] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[15] Daniel M. Bikel, Richard M. Schwartz, and Ralph M. Weischedel. An algorithm that learns what's in a name. *Machine Learning*, 34:211–231, 2004.

[16] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomás Mikolov. Enriching word vectors with subword information. *CoRR*, abs/1607.04606, 2016.

[17] Andrew Eliot Borthwick. *A Maximum Entropy Approach to Named Entity Recognition.* PhD thesis, USA, 1999. AAI9945252.

[18] Sergey Brin. Extracting patterns and relations from the world wide web. In *In WebDB Workshop at 6th International Conference on Extending Database Technology, EDBT'98*, pages 172–183, 1998.

[19] Xavier Carreras, Lluís Màrquez, and Lluís Padró. Named entity extraction using adaboost. pages 1–4, 08 2002.

[20] Casimiro Pio Carrino, Jordi Armengol-Estapé, Ona de Gibert Bonet, Asier Gutiérrez-Fandiño, Aitor Gonzalez-Agirre, Martin Krallinger, and Marta Villegas. Spanish biomedical crawled corpus: A large, diverse dataset for spanish biomedical language models, 2021.

[21] Michael Collins and Yoram Singer. Unsupervised models for named entity classification. In *1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, 1999.

[22] Ronan Collobert, Jason Weston, Leon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12:2493–2537, 02 2011.

[23] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Mach. Learn.*, 20(3):273–297, sep 1995.

[24] Leyang Cui and Yue Zhang. Hierarchically-refined label attention network for sequence labeling. pages 4106–4119, 01 2019.

[25] James R. Curran and Stephen Clark. Language independent ner using a maximum entropy tagger. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003 - Volume 4*, CONLL '03, page 164–167, USA, 2003. Association for Computational Linguistics.

[26] Joaquim F. Ferreira da Silva, Zornitsa Kozareva, and José Gabriel Pereira Lopes. Cluster analysis and classification of named entities. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, Lisbon, Portugal, May 2004. European Language Resources Association (ELRA).

[27] Allan Peter Davis, Cynthia J Grondin, Robin J Johnson, Daniela Sciaky, Jolene Wiegers, Thomas C Wiegers, and Carolyn J Mattingly. Comparative Toxicogenomics Database (CTD): update 2021. *Nucleic Acids Research*, 49(D1):D1138–D1143, 10 2020.

[28] Franck Dernoncourt, Ji Young Lee, and Peter Szolovits. NeuroNER: an easy-to-use program for named-entity recognition based on neural networks. *Conference on Empirical Methods on Natural Language Processing (EMNLP)*, 2017.

[29] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[30] Rezarta Dogan, Robert Leaman, Sun Kim, Dongseop Kwon, Chih-Hsuan Wei, Donald Comeau, Yifan Peng, David Cissel, Cathleen Coss, Carol Fisher, Rob Guzman, Preeti Kochar, Stella Koppel, Dorothy Trinh, Keiko Sekiya, Janice Ward, Deborah Whitman, Susan Schmidt, and Zhiyong lu. Nlm-chem, a new resource for chemical entity recognition in pubmed full text literature. *Scientific Data*, 8, 03 2021.

[31] Rezarta Dogan, Robert Leaman, and Zhiyong lu. Ncbi disease corpus: A resource for disease name recognition and concept normalization. *Journal of biomedical informatics*, 47, 01 2014.

[32] Salim Dridi. Unsupervised learning - a systematic literature review, 12 2021.

[33] Oren Etzioni, Michael Cafarella, Doug Downey, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel S. Weld, and Alexander Yates. Unsupervised named-entity extraction from the web: An experimental study. *Artificial Intelligence*, 165(1):91–134, 2005.

[34] Richard Evans. A framework for named entity recognition in the open domain. *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP), 2003*, 01 2004.

[35] G.D. Forney. The viterbi algorithm. *Proceedings of the IEEE*, 61(3):268–278, 1973.

[36] Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke S. Zettlemoyer. Allennlp: A deep semantic natural language processing platform. 2017.

[37] Ralph Grishman and Beth Sundheim. Design of the muc-6 evaluation. In *Proceedings of the 6th Conference on Message Understanding*, MUC6 '95, page 1–11, USA, 1995. Association for Computational Linguistics.

[38] Jiafeng Guo, Gu Xu, Xueqi Cheng, and Hang Li. Named entity recognition in query. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '09, page 267–274, New York, NY, USA, 2009. Association for Computing Machinery.

[39] J. M. Hammersley and P. E. Clifford. Markov random fields on finite graphs and lattices. Unpublished manuscript, 1971.

[40] Marti A. Hearst. Automatic acquisition of hyponyms from large text corpora. In *COLING 1992 Volume 2: The 14th International Conference on Computational Linguistics*, 1992.

[41] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9:1735–80, 12 1997.

[42] Zhiheng Huang, Wei Xu, and Kai Yu. Bidirectional lstm-crf models for sequence tagging, 2015.

[43] Oliver C. Ibe. Chapter 12 - special random processes. In Oliver C. Ibe, editor, *Fundamentals of Applied Probability and Random Processes (Second Edition)*, pages 369–425. Academic Press, Boston, second edition edition, 2014.

[44] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomás Mikolov. Bag of tricks for efficient text classification. *CoRR*, abs/1607.01759, 2016.

[45] Arzoo Katiyar and Claire Cardie. Nested named entity recognition revisited. 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 861–871, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[46] Sun Kim, Lana Yeganova, Donald Comeau, W. Wilbur, and Zhiyong lu. Pubmed phrases, an open set of coherent phrases for searching biomedical literature. *Scientific Data*, 5:180104, 06 2018.

[47] Onur Kuru, Ozan Arkan Can, and Deniz Yuret. CharNER: Character-level named entity recognition. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 911–921, Osaka, Japan, December 2016. The COLING 2016 Organizing Committee.

[48] John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, page 282–289, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.

[49] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270, San Diego, California, June 2016. Association for Computational Linguistics.

[50] Yann LeCun, Y. Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521:436–44, 05 2015.

[51] Yann Lecun, Leon Bottou, Y. Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86:2278 – 2324, 12 1998.

[52] Gang Li, Karen E. Ross, Cecilia N. Arighi, Yifan Peng, Cathy H. Wu, and K. Vijay-Shanker. mirtex: A text mining system for mirna-gene relation extraction. *PLOS Computational Biology*, 11(9):1–24, 09 2015.

[53] Jing Li, Aixin Sun, Jianglei Han, and Chenliang Li. A survey on deep learning for named entity recognition. *CoRR*, abs/1812.09449, 2018.

[54] Peng-Hsuan Li, Ruo-Ping Dong, Yu-Siang Wang, Ju-Chieh Chou, and Wei-Yun Ma. Leveraging linguistic structures for named entity recognition with bidirectional recursive neural networks. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2664–2669, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

[55] Xiaoya Li, Jingrong Feng, Yuxian Meng, Qinghong Han, Fei Wu, and Jiwei Li. A unified MRC framework for named entity recognition. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5849–5859, Online, July 2020. Association for Computational Linguistics.

[56] Bill Y. Lin, Frank Xu, Zhiyi Luo, and Kenny Zhu. Multi-channel BiLSTM-CRF model for emerging named entity recognition in social media. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pages 160–165, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

[57] Tianyu Liu, Jin-Ge Yao, and Chin-Yew Lin. Towards improving neural named entity recognition with gazetteers. In *Proceedings of the 57th Annual Meeting of the Asso-*

ciation for Computational Linguistics, pages 5301–5307, Florence, Italy, July 2019. Association for Computational Linguistics.

[58] Xuezhe Ma and Eduard Hovy. End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1064–1074, Berlin, Germany, August 2016. Association for Computational Linguistics.

[59] Andrzej Maćkiewicz and Waldemar Ratajczak. Principal components analysis (pca). *Computers & Geosciences*, 19:303–342, 1993.

[60] Christopher Manning, Mihai Surdeanu, John Bauer, Jenny 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, Baltimore, Maryland, June 2014. Association for Computational Linguistics.

[61] Andrew McCallum and Wei Li. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 188–191, 2003.

[62] Paul McNamee and James Mayfield. Entity extraction without language-specific resources. In *Proceedings of the 6th Conference on Natural Language Learning - Volume 20*, COLING-02, page 1–4, USA, 2002. Association for Computational Linguistics.

[63] Jianyu Miao and Lingfeng Niu. A survey on feature selection. *Procedia Computer Science*, 91:919–926, 12 2016.

[64] Andrei Mikheev, Marc Moens, and Claire Grover. Named entity recognition without gazetteers. In *Proceedings of the Ninth Conference on European Chapter of the Association for Computational Linguistics*, EACL '99, page 1–8, USA, 1999. Association for Computational Linguistics.

[65] Tomas Mikolov, G.s Corrado, Kai Chen, and Jeffrey Dean. Efficient estimation of word representations in vector space. pages 1–12, 01 2013.

[66] Sunil Mohan and Donghui Li. Medmentions: A large biomedical corpus annotated with umls concepts, 2019.

[67] Diego Mollá, Menno van Zaanen, and Daniel Smith. Named entity recognition for question answering. In *Proceedings of the Australasian Language Technology Workshop 2006*, pages 51–58, Sydney, Australia, November 2006.

[68] Seungwhan Moon, Leonardo Neves, and Vitor R. Carvalho. Multimodal named entity recognition for short social media posts. In *NAACL*, 2018.

[69] Robert Munro and Christopher D. Manning. Accurate unsupervised joint named-entity extraction from unaligned parallel text. In *Proceedings of the 4th Named Entity Workshop*, NEWS '12, page 21–29, USA, 2012. Association for Computational Linguistics.

[70] David Nadeau and Satoshi Sekine. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30, 08 2007.

[71] Mary Ellen Okurowski, Harold Wilson, Joaquin Urbina, Tony Taylor, Ruth Colvin Clark, and Frank Krapcho. Text summarizer in use: Lessons learned from real world deployment and evaluation. In *Proceedings of the 2000 NAACL-ANLPWorkshop on Automatic Summarization - Volume 4*, NAACL-ANLP-AutoSum '00, page 49–58, USA, 2000. Association for Computational Linguistics.

[72] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

[73] Matthew E. Peters, Waleed Ammar, Chandra Bhagavatula, and Russell Power. Semi-supervised sequence tagging with bidirectional language models. In *ACL*, 2017.

[74] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[75] T. Poibeau and Leila Kosseim. Proper name extraction from non-journalistic texts. In *CLIN*, 2000.

[76] Alec Radford and Karthik Narasimhan. Improving language understanding by generative pre-training. 2018.

[77] L.F. Rau. Extracting company names from text. In *[1991] Proceedings. The Seventh IEEE Conference on Artificial Intelligence Application*, volume i, pages 29–32, 1991.

[78] Marek Rei. Semi-supervised multitask learning for sequence labeling. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2121–2130, Vancouver, Canada, July 2017. Association for Computational Linguistics.

[79] Marek Rei, Gamal Crichton, and Sampo Pyysalo. Attending to characters in neural sequence labeling models. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 309–318, Osaka, Japan, December 2016. The COLING 2016 Organizing Committee.

[80] Alan Ritter, Sam Clark, Mausam, and Oren Etzioni. Named entity recognition in tweets: An experimental study. In *EMNLP*, 2011.

[81] Alan Ritter, Mausam, Oren Etzioni, and Sam Clark. Open domain event extraction from twitter. In *KDD*, 2012.

[82] Yanyao Shen, Hyokun Yun, Zachary C. Lipton, Yakov Kronrod, and Animashree Anandkumar. Deep active learning for named entity recognition, 2018.

[83] Emma Strubell, Patrick Verga, David Belanger, and Andrew McCallum. Fast and accurate entity recognition with iterated dilated convolutions. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2670–2680, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

[84] Christine Thielen. An approach to proper name tagging for german. *arXiv: Computation and Language*, 1995.

[85] Quan Tran, Andrew MacKinlay, and Antonio Jimeno Yepes. Named entity recognition with stack residual LSTM and trainable bias decoding. In *Proceedings of the Eighth*

International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 566–575, Taipei, Taiwan, November 2017. Asian Federation of Natural Language Processing.

[86] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[87] Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. Pointer networks. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015.

[88] Lucy Lu Wang, Kyle Lo, Yoganand Chandrasekhar, Russell Reas, Jiangjiang Yang, Doug Burdick, Darrin Eide, Kathryn Funk, Yannis Katsis, Rodney Kinney, Yunyao Li, Ziyang Liu, William Merrill, Paul Mooney, Dewey Murdick, Devvret Rishi, Jerry Sheehan, Zhihong Shen, Brandon Stilson, Alex Wade, Kuansan Wang, Nancy Xin Ru Wang, Chris Wilhelm, Boya Xie, Douglas Raymond, Daniel S. Weld, Oren Etzioni, and Sebastian Kohlmeier. Cord-19: The covid-19 open research dataset, 2020.

[89] Xinglong Wang, Jun'ichi Tsujii, and Sophia Ananiadou. Disambiguating the species of biomedical named entities using natural language parsers. *Bioinformatics (Oxford, England)*, 26:661–7, 03 2010.

[90] Qikang Wei, Tao Chen, Ruifeng Xu, and Lin Gui. Disease named entity recognition by combining conditional random fields and bidirectional recurrent neural networks. *Database: The Journal of Biological Databases and Curation*, 2016, 10 2016.

[91] Janet Wiles and Jeffrey Elman. Learning to count without a counter: A case study of dynamics and activation landscapes in recurrent networks. 06 1995.

[92] L. Wolf and Amnon Shashua. Feature selection for unsupervised and supervised inference: The emergence of sparsity in a weight-based approach. volume 6, pages 378 – 384 vol.1, 11 2003.

[93] Alejandro Piad-Morfis y Suilan Estevez-Velarde y Yoan Gutierrez y Yudivian Almeida-Cruz y Andrés Montoyo y Rafael Muñoz. Overview of the ehealth knowledge discovery challenge at iberlef 2021. *Procesamiento del Lenguaje Natural*, 67(0):233–242, 2021.

[94] Jie Yang, Yue Zhang, and Fei Dong. Neural reranking for named entity recognition. pages 784–792, 11 2017.

[95] Roman Yangarber, Winston Lin, and Ralph Grishman. Unsupervised learning of generalized names. In *Proceedings of the 19th International Conference on Computational Linguistics - Volume 1*, COLING '02, page 1–7, USA, 2002. Association for Computational Linguistics.

[96] Lin Yao, Hong Liu, Yi Liu, Xinxin Li, and Muhammad Waqas Anwar. Biomedical named entity recognition based on deep neutral network. *International Journal of Hybrid Information Technology*, 8:279–288, 2015.

[97] Zhixiu Ye and Zhen-Hua Ling. Hybrid semi-Markov CRF for neural sequence labeling. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 235–240, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[98] Feifei Zhai, Saloni Potdar, Bing Xiang, and Bowen Zhou. Neural models for sequence chunking. 01 2017.

[99] Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. ERNIE: Enhanced language representation with informative entities. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1441–1451, Florence, Italy, July 2019. Association for Computational Linguistics.

[100] Zheng Zhao and Huan Liu. Semi-supervised feature selection via spectral analysis. In *SDM*, 2007.

[101] Suncong Zheng, Feng Wang, Hongyun Bao, Yuexing Hao, Peng Zhou, and Bo Xu. Joint extraction of entities and relations based on a novel tagging scheme. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*

(Volume 1: Long Papers), pages 1227–1236, Vancouver, Canada, July 2017. Association for Computational Linguistics.

[102] GuoDong Zhou and Jian Su. Named entity recognition using an hmm-based chunk tagger. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, page 473–480, USA, 2002. Association for Computational Linguistics.

[103] Peng Zhou, Suncong Zheng, Jiaming Xu, Zhenyu Qi, Hongyun Bao, and Bo Xu. Joint extraction of multiple relations and entities by using a hybrid neural network. pages 135–146, 10 2017.

[104] Xuezhong Zhou, Jörg Menche, Albert-Laszlo Barabasi, and Amitabh Sharma. Human symptoms–disease network. *Nature communications*, 5:4212, 06 2014.

[105] Xiaojin Jerry Zhu. Semi-supervised learning literature survey. 2005.