Pontifícia Universidade Católica
do Rio de Janeiro

Lucas Aguiar Pavanelli

An End-to-End Model for Joint Entity and Relation Extraction in Portuguese

Dissertação de Mestrado

Dissertation presented to the Programa de Pós–graduação em Informática of PUC-Rio in partial fulfillment of the requirements for the degree of Mestre em Informática.

Advisor: Prof. Eduardo Sany Laber

Rio de Janeiro
September 2022

Pontifícia Universidade Católica
do Rio de Janeiro

Lucas Aguiar Pavanelli

An End-to-End Model for Joint Entity and Relation Extraction in Portuguese

Dissertation presented to the Programa de Pós–graduação em Informática of PUC-Rio in partial fulfillment of the requirements for the degree of Mestre em Informática. Approved by the Examination Committee:

Prof. Eduardo Sany Laber
Advisor
Departamento de Informática – PUC-Rio

Prof. Sergio Colcher
Departamento de Informática – PUC-Rio

Dr. Thiago Castro Ferreira
UFMG

Rio de Janeiro, September 5th, 2022

Lucas Aguiar Pavanelli

Majored in Computer Engineering by PUC-Rio.

Bibliographic data

Pavanelli, Lucas Aguiar

An End-to-End Model for Joint Entity and Relation Extraction in Portuguese / Lucas Aguiar Pavanelli; advisor: Eduardo Sany Laber. – 2022.

58 f: il. color. ; 30 cm

Dissertação (mestrado) - Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Informática, 2022.

Inclui bibliografia

1. Departamento de Informática – Teses. 2. Ciência da Computação – Teses. 3. Processamento de Linguagem Natural. 4. Reconhecimento de Entidades Nomeadas. 5. Extração de Relações Semânticas. 6. Aprendizagem Profunda. I. Laber, Eduardo. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Informática. III. Título.

CDD: 004

To my parents, for their support
and encouragement.

Acknowledgments

I would like to thank my advisor Eduardo Laber for the help in this journey. Also, I want to thank my family and friends for all the support when things were tough.

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

Abstract

Pavanelli, Lucas Aguiar; Laber, Eduardo (Advisor). **An End-to-End Model for Joint Entity and Relation Extraction in Portuguese**. Rio de Janeiro, 2022. 58p. Dissertação de Mestrado – Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

Natural language processing (NLP) techniques are becoming popular recently. The range of applications that benefit from NLP is extensive, from building machine translation systems to helping market a product. Within NLP, the Information Extraction (IE) field is widespread; it focuses on processing texts to retrieve specific information about a particular entity or concept. Still, the research community mainly focuses on building models for English data. This thesis addresses three tasks in the IE domain: Named Entity Recognition, Relation Extraction, and Joint Entity and Relation Extraction. First, we created a novel Portuguese dataset in the biomedical domain, described the annotation process, and measured its properties. Also, we developed a novel model for the Joint Entity and Relation Extraction task, verifying that it is competitive compared to other models. Finally, we carefully evaluated proposed models on non-English language datasets and confirmed the dominance of neural-based models.

Keywords

Natural Language Processing; Named Entity Recognition; Relation Extraction; Deep Learning.

Resumo

Pavanelli, Lucas Aguiar; Laber, Eduardo. **Modelo end-to-end para Extração de Entidades e Relações de forma conjunta em Português**. Rio de Janeiro, 2022. 58p. Dissertação de Mestrado – Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

As técnicas de processamento de linguagem natural (NLP) estão se tornando populares recentemente. A gama de aplicativos que se beneficiam de NLP é extensa, desde criar sistemas de tradução automática até ajudar no marketing de um produto. Dentro de NLP, o campo de Extração de Informações (IE) é difundido; concentra-se no processamento de textos para recuperar informações específicas sobre uma determinada entidade ou conceito. Ainda assim, a comunidade de pesquisa se concentra principalmente na construção de modelos para dados na língua inglesa. Esta tese aborda três tarefas no domínio do IE: Reconhecimento de Entidade Nomeada, Extração de Relações Semânticas e Extração Conjunta de Entidade e Relação. Primeiro, criamos um novo conjunto de dados em português no domínio biomédico, descrevemos o processo de anotação e medimos suas propriedades. Além disso, desenvolvemos um novo modelo para a tarefa de Extração Conjunta de Entidade e Relação, verificando que o mesmo é competitivo em comparação com outros modelos. Finalmente, avaliamos cuidadosamente os modelos propostos em textos de idiomas diferentes do inglês e confirmamos a dominância de modelos baseados em redes neurais.

Palavras-chave

Processamento de Linguagem Natural; Reconhecimento de Entidades Nomeadas; Extração de Relações Semânticas; Aprendizagem Profunda.

Table of contents

List of figures

List of tables

List of Abreviations

ML – Machine Learning
NLP – Natural Language Processing
IE – Information Extraction
NER – Named Entity Recognition
RE – Relation Extraction
CRF – Conditional Random Fields
SVM – Support Vector Machine

1
Introduction

Information Extraction is a popular Natural Language Processing (NLP) field with many real-world applications: search engines [64, 65, 66], recommender systems [67, 68, 69], and document classification [70, 71, 72]. Named Entity Recognition (NER) and Relation Extraction (RE) are central tasks inside Information Extraction. The former is concerned with identifying portions of a text (entities) that convey a specific meaning, for example, detecting a place's name. The latter is concerned with the relations between the entities, for example, spotting that Rio de Janeiro **is a** city of Brazil.

Much work has been done regarding NER and RE [73, 74]. In the beginning, rule-based or dictionary-based methods were used to recognize entities and relations. Developing those models does not require much computation or complex code, but they require specific domain knowledge and can not be applied to a more general scenario. With the rise of neural networks, these were also applied to NER and RE, causing better results and pushing the field further.

With the rise of deep learning methods, researchers considered solving NER and RE tasks jointly, naming it Joint Entity and Relation Extraction. If the end goal is to detect the relation from a text, we might solve both NER and RE with a single model. That way, information from the NER portion of the model can also be used for the RE part.

Apart from the models, data is the primary concern of researchers and engineers working on Machine Learning. Since most of the community are English speakers, datasets are usually built-in English language. However, that does not translate to our world. There are more than 7,000 languages globally, and English speakers are 20% of the world population [63]. Therefore, the need for data on other languages is severe.

This thesis addresses the mentioned tasks: Named Entity Recognition (NER), Relation Extraction (RE), and Joint Entity and Relation Extraction. For NER and RE, we implemented different models, from baseline ML models to up-to-date deep learning ones. For Joint Entity and Relation Extraction, we developed a new deep learning model called the Joint model.

Also, we built a new dataset on Portuguese text from the medical domain

called Bete. We collected answers from medical students to diabetes questions from a general audience. We guided the annotations of such answers regarding entities and relationships between them. Finally, we calculated statistics and performed the first evaluation of the data.

Finally, we extensively evaluated the models on the Bete dataset and another dataset containing English and Spanish sentences, called eHealth-KD. We evaluated standalone NER and RE models and also compared the Joint model with other systems on the eHealth-KD dataset.

In order for others to use the novel dataset, the Joint model, and for reproducibility, we open-sourced the code: Multilingual Entity And Relation Extraction Evaluation - GitHub repository.

1.1 Research Questions

To make our focus clear, here we delineate research questions that we aim to answer:

1. What is the models' performance concerning languages different than English?

2. How do NER and RE models perform on the new Bete corpus?

3. How can we create a model that jointly solves NER and RE tasks?

4. How does relation extraction benefit from entity recognition?

The first two questions concern the new Bete dataset and the evaluation of data not in English. Since the novel dataset is now available, we want to know how popular methods perform on the data. Also, to enrich the community perspective on non-English text, we want to answer how models perform on Portuguese and Spanish data.

The last two questions are related to the new Joint Entity and Relation Extraction model. The aim is to investigate the performance of the novel model. Also, we want to investigate how Relation Extraction benefits from Named Entity Recognition in the neural network.

1.2 Contributions

This thesis introduces a novel model for Joint Entity and Relation Extraction. Moreover, we perform an extensive evaluation of information extraction tasks. Here we list the following as main contributions:

- Development of a new Portuguese dataset for the Named Entity Recognition and Relation Extraction tasks.
- Evaluation of NER, RE, and Joint model on non-English datasets.
- Development of a new Joint Entity and Relation Extraction model.
- Results analysis and comparison of standalone NER and RE models and a Joint Entity and Relation Extraction model on non-English data.

We emphasize that this work focuses on non-English data by building a novel dataset and evaluating Portuguese and Spanish data. This contribution is significant because it fosters a more democratic usage of AI by providing products for non-English speakers.

1.3 Outline

This thesis is composed of six chapters.

Chapter 2 provides the necessary background. It introduces the machine learning techniques, NER, RE, and Joint Entity Extraction tasks and provides the related work.

Chapter 3 describes the used data. We inform what the dataset is about and give statistics about each dataset.

Chapter 4 explains what models we used and built.

Chapter 5 clarifies our experimental setup.

Chapter 6 describes our experiments and discussion.

Chapter 7 summarizes what we did and opens the path for future work.

2
Background and Related Work

This section gives the necessary background to follow the rest of the thesis. Also, we summarize the research works done, walking through the history of methods used to address the NER and RE tasks. We divided this chapter into two parts. The first describes the Named Entity Recognition task, and Relation Extraction is in the second part.

2.1
Named Entity Recognition

Named Entity Recognition, also called NER, is an information extraction task that aims to identify significant parts of a text and classify them, such as Person, Organization, and Location. Figure 2.1 shows an example. Given the sentence: “Tom Jobim is a musician from Rio de Janeiro.” we can identify two entities: "Tom Jobin" as Person and "Rio de Janeiro" as Location.

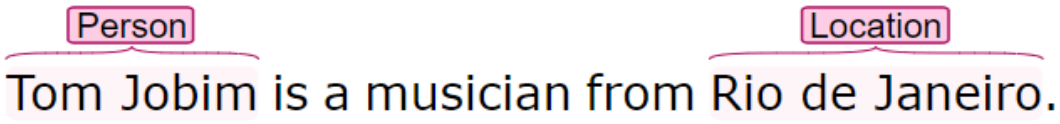

Figure 2.1: Named entity recognition example.

The importance of NER relies on the fact that getting structured information from text is fundamental for some applications. For example, we could use it in downstream tasks such as chatbots, question answering, and knowledge graph construction.

A usual way to represent entities is by using the IOB2 format. First, we need to tokenize the text. Then, we give a different symbol for tokens at the beginning of an entity (“B-”) and tokens in the middle of an entity (“I-”). For the same example, Figure 2.2 illustrates the tokens and respective entities in IOB2 format. Here we used “O” to refer to empty entities.

The following sections outline the history of NER. From rule-based approaches to nowadays neural methods.

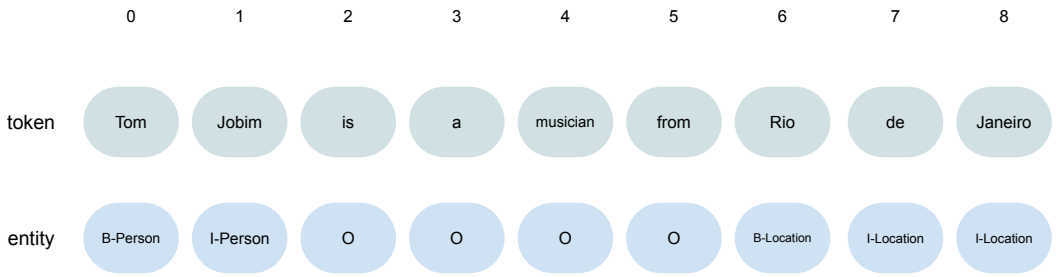

Figure 2.2: IOB2 example.

2.1.1 Rule-based

This approach relies on hand-crafted rules to identify critical pieces of the text. The rules can be based on regular expression, pre-processed dictionary, morphological analysis, semantic and syntactic rules, and hand-crafted grammar.

In the biomedical domain, Fukuda et al. [6] introduce a method called PROPER that uses the characteristics of proper nouns to identify proteins in medical and biological documents. Furthermore, Gaizauskas et al. [7] built the PASTA system that uses regular expressions and morphological analysis to fill templates and extract protein structure information from scientific articles. Finally, Hanisch et al. [8] proposed ProMiner, a system that detects entities using a pre-processed dictionary with the biological entities and all known synonyms.

One of the main advantages of rule-based systems is that they do not need training data. Languages that do not have many resources are good examples where data is not available; hence rule-based rules can be used. For example, Farmakiotou et al. [9] proposed a system based on dictionaries and hand-crafted lexical rules to identify named entities in Greek. Küçük and Yazici [10] built a NER system for Turkish. They used lexical resources, such as a dictionary of person names and a list of well-known locations, and created patterns to extract named entities. Riaz [11] discussed the challenges of tackling NER in languages that have small annotated corpora and proposed a rule-based method for Urdu based on lexical cues.

When we port rule-based systems to a different domain from the one they were created, their performance usually decreases. This is because these rules are particular to the domain they were created; it is challenging to tackle a general scenario. Because of that, we have rule-based systems devised for

diverse domains. For example, Eftimov et al. [12] proposed a method to extract dietary information, such as food, nutrient, and quantity. Popovski et al. [13] proposed FoodIE, a method that involves rules based on part-of-speech (POS) and semantics tags, to identify food named entities.

2.1.2 Feature-based Supervised Methods

Feature-based NER systems rely on extracting features from the training data. These features are then fed to a machine learning algorithm that should be able to generalize to unseen data. By applying this approach, NER can be seen as a sequence label task, i.e., for each unit, choose the corresponding tag, or as a multi-label classification problem, i.e., given the features of an input sentence, return all recognized entities.

Standard features can be divided into two groups: 1) word-level features such as if a word is a punctuation mark, a number, or if it has uppercase characters. 2) document features such as the number of occurrences of each entity type or the majority label assigned for a token.

As for the supervised methods used, popular ones are hidden markov model (HMM) [14], support vector machine (SVM) [15], decision trees [16], and conditional random field (CRF) [17].

Zhou and Su [18] proposed a method using HMM that is fed with word-level features, such as capitalization and digitalization, and external macro context feature that looks in the list of already recognized entities.

Isozaki and Kazawa [19] showed that SVM outperformed previous rule-based systems and also proposed a method to make the SVM-based NER faster.

Carreras et al. [20] presented a method for the CoNLL 2002 competition that used binary AdaBoost [21] classifiers. They used word-level features such as ortographic (if it is capitalized or a digit), part-of-speech, and bag-of-words.

Krishnan and Manning [22] proposed an approach based on two CRF: the first used local features, and the second used local information and features from the output of the first one. Moreover, they made it easy to incorporate non-local features by changing the output of the first CRF.

Darwish [59] introduced cross-lingual links between English and Arabic to achieve top performance in Arabic. The work exploited English's orthographic features as well as Arabic and English Wikipedias, including annotations from significant knowledge sources.

2.1.3 Neural Methods

With the advance of AI, neural-based methods have become ubiquitous in the research community. One of the core strengths is automatically discovering complex features without needing to specify them.

Collobert et al. [52] proposed one of the first neural network architectures for the NER task. They built feature vectors using orthographic features and applied a convolutional neural network (CNN). Some years later, Collobert et al. [53] improved the feature vector representation by using word embeddings: N-dimensional vectors representing words.

Many works also used word embeddings, Yao et al. [54] trained word-level representations on the PubMed database using the skip-gram model [55] and created a neural biomedical NER model. Nguyen et al. [56] proposed a recurrent neural network (RNN) architecture system using word embeddings trained on English text from Gigaword corpus.

Apart from representing words as vectors, other approaches used character-level embeddings. One advantage is that we can represent out-of-vocabulary words by inferring from the characters. Ma et al. [57] extracted both word and character-level representations and fed them to a RNN context encoder. Kuru et al. [58] proposed CharNER: a bidirectional long short-term memory (LSTM) network that outputs tag probabilities for each character. These probabilities are used to infer word-level tags by applying the Viterbi decoder. They show their performance in seven languages and conclude that using character as the primary representation is superior to words as the basic unit.

2.2 Relation Extraction

Relation extraction (RE) is also a task in the Information Extraction (IE) domain. It consists of predicting what predefined relation exists between two mentioned entities in the text or if there is not a relationship. Figure 2.3 illustrates a relationship: the entity “Tom Jobim” relates to “Rio de Janeiro” by the relationship “is-from”.

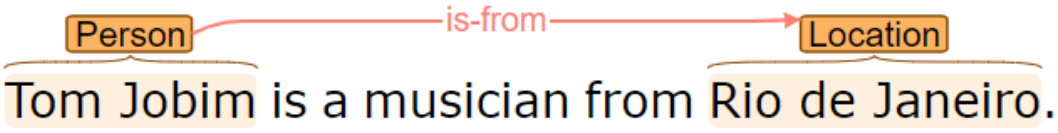

Figure 2.3: Relation extraction example.

As in the case of NER, Relation Extraction is relevant for downstream tasks, as it provides more complex structured information about the text.

The evolution of RE follows a similar path to NER, as it evolves from feature-based machine learning approaches to neural methods.

2.2.1 Feature-based

A feature-based approach for RE requires labeled data in the following format: for each relation instance (pair of entity mentions), assign a relation label or NONE if there is no label. First, features can be generated from the relation instances; then, a classifier is used to predict the label of a new relation.

Features can be related to the entity mentions, for example, which words and POS tag constitute each entity's mentions or tag. Also, we can have features related to the context of each mentioned entity, for example, how many words or other entities are between the mentioned entities.

Kambhatla [23] used lexical, syntactic, and semantic features and Maximum Entropy models to extract relations. They extracted the features described above and the dependencies and parse tree connecting the two mentioned entities.

Zhou et al. [24] also proposed using lexical, syntactic, and semantic features, but the authors applied an SVM to classify the relations. They showed that the base phrase chunking information is efficient for relation extraction. This type of feature provides the model with the phrase heads in the context of the entity mentions, for example, the phrase heads between the two mentions or the first and second phrase heads before the mentions.

Jiang and Zhai [25] systematically evaluated what features are adequate for the relation extraction task. They experimented with three feature subspaces: sequence, syntactic parse tree, and dependency parse tree. Results showed that each subspace is effective, and combining them did not improve the results considerably.

2.2.2 Kernel-based

Another approach to relation extraction is to create a kernel to identify the positives and negatives relation instances. This approach does not require feature engineering as the feature-based ones. The kernel methods can use a variety of properties from the words, such as syntactic tree and POS tags.

Zelenko et al. [26] proposed kernels defined over representations of crucial elements from the text and used Support Vector Machine and Voted Perceptron

learning algorithms.

Bunescu et al. [27] presented a kernel method based on a generalization of subsequence kernels. This work uses three subsequence patterns normally employed to assert a relationship between two entities.

2.2.3 Neural methods

As for NER, neural-based models have state-of-the-art performance in the Relation Extraction task.

Liu et al. [60] introduced one of the first neural network architectures to learn features instead of devising hand-crafted features automatically. Also, they incorporated some lexical features, e.g. POS tags and entity types. The work proposed an end-to-end system using CNNs and showed improved results over kernel-based methods, opening the path to novel neural-based approaches.

Zeng et al. [61] employed word embeddings trained using an unsupervised approach on a large corpus. The work concatenated the embeddings with lexical features to generate a representation from a sentence and applied a softmax classifier to predict the relationship.

Nguyen et al. [62] removed the dependencies on lexical features by learning the features using only a CNN. In addition, they used multiple window sizes for filters and fine-tuned pre-trained word embeddings.

3
Data

This section introduces the used datasets. We describe Bete and eHealth-KD: how they were created, what type of text they have, and what entities and relationships were annotated.

3.1
Bete

Bete is a novel annotated Brazilian Portuguese Named Entity Recognition and Relation Extraction corpus in the Diabetes Mellitus domain. We created this dataset to develop a framework to automatically identify diabetes-related entities and relations in texts produced by medical students as answers to user queries in diabetes-related public forums. Considering the low number of studies regarding texts in the medical domain targeting the general public and studies focusing on Brazilian Portuguese, our study is valuable to both contexts.

In this thesis, we contributed to creating the dataset by collecting answers from medical students, guiding named entities and relationship annotations by students, calculating data statistics, and performing the first evaluation using up-to-date models.

Next, we cover all the details of Bete's development.

3.1.1
Annotation

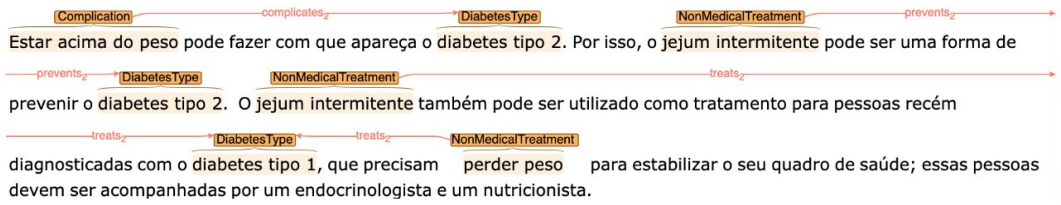

Figure 3.1: Annotation example from our corpus.

Annotation Setup As an annotation tool, we used Webanno [28], an open-source and intuitive software/platform. After comparing several available entity tagging tools, we found Webanno was the easiest and most efficient tool for our purposes. The system was set up on a web server; data was uploaded for each user and entity types defined within the system.

Annotation Guidelines The annotation guidelines were created in an iterative process. A first draft was created containing general guidelines and specific examples of types of entities. Domain specialists were consulted regarding annotators' queries, and their answer was then used to update the guidelines, after which they were tested again. Besides, during the annotation process itself, whenever one of the annotators ran into an unclear situation, this was added as an example to the guidelines. Figure 3.1 shows an example of a text annotated in Webanno following our guidelines [1].

Entity Types The set of entity types devised for our annotation was built drawing on an ontology proposed in [29]. Table 3.1 lists the targeted entities and provides a brief explanation of each type with some examples.

Annotation Process We recruited undergraduate students pursuing their BA degrees to complete the annotation task. The students are part of Empoder@, a multidisciplinary project with health sciences, statistics, computer sciences, and applied linguistics students. Based on the institutional exchange, they develop research and human resource training aiming to empower researchers, professionals, and users of the Health Service.

The annotators received a training session and were asked to read the guidelines and resort to the project coordinator whenever they encountered problems during annotation. Two students annotated each document, and a third one reviewed the work.

3.1.2 Dataset Information

The dataset comprises two sets of documents composed of answers to diabetes-related general questions: a first set containing 304 documents drafted by medical students under the supervision of medical professionals and a

[1]Translation into English: "Being overweight can lead to type 2 diabetes. Therefore, intermittent fasting may be a way to prevent type 2 diabetes. Intermittent fasting can also be used as a treatment for people newly diagnosed with type 1 diabetes who need to lose weight to achieve a more stable health condition; these people should be advised and monitored by an endocrinologist and a nutritionist."

second one containing 201 documents written by nutritional science students supervised by professionals. Table 3.2 shows overall statistics of the whole dataset.o

Table 3.3 shows the number of annotations per entity type, while table 3.4 covers the annotated relations.

3.2 eHealth-KD

The eHealth-KD dataset is from the IberLEF eHealth Knowledge Discovery Challenge 2021 [31], a challenge that targets the recognition of entities and their relations in the clinical domain, encouraging researchers and scientists to discover new knowledge through text mining and NLP in the health domain. It consists mainly of electronic health documents in Spanish. The challenge is to extract structure from the text by identifying four types of general-purpose entities and thirteen semantic relations. Also, English text and sentences from different domains were added to the dataset to make it multilingual and multi-domain and incentivize models with these characteristics.

Figure 3.2 shows examples of entities and relations that compose the dataset. Table 3.5 shows entities descriptions. Table 3.6 shows relations descriptions.

Since the dataset comes from a research challenge, the organizers provide already annotated data. They give Brat-annotated [30] files and also Python scripts to parse the files into entities and relations for each sentence.

Because the input is tokenized by spaces and the entities/relations are linked to each token, we need to parse the input file according to the used models. For example, we developed a script to tokenize the data given a BERT model. Since BERT models tokenize the sentences differently, we need to align the entity/relation annotations and the tokens.

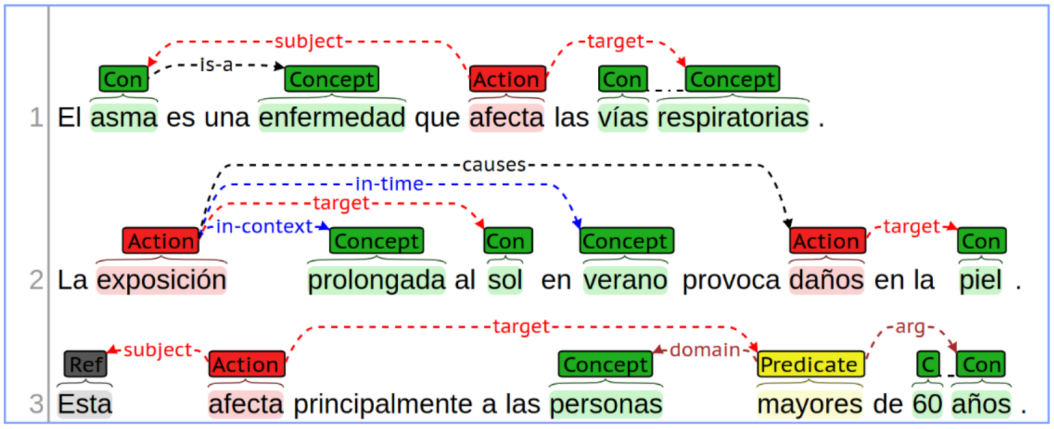

Figure 3.2: Annotation of Spanish sentences from eHealth-KD dataset. For example, “asma” (asthma) was annotated as Concept entity and “enfermedad” (illness) as Concept as well. Also, there is a relation “is-a” between those two entities, which means asthma is an instance of illness. Extracted from [31].

Category	Description	Examples of annotated entities
Diabetes Type	subclass of diabetes	type 2 – type 1
Complication	diseases and health conditions causing or caused by diabetes	being overweight – wounds – depression - neuropathy
Symptom	physical or mental condition experienced by the patient regarded as indicating diabetes	low blood sugar
Glucose Value	measurement of blood sugar level	250 – 100 - 80
Insulin	insulin type	NPH – Aspart
Medication	prescribed drugs or medicine	Metformin - Tetracaine hydrochloride
Non Medical Treatment	healthcare activities or behavior other than prescribed medication	intermittent fasting – physical exercise
Food	source of nutritional support for organisms	peanut butter, candies, bread
Dose	amount, quantity or size	150ml – 200g – 1 glass
Test	medical exams	blood test - glycosylated hemoglobin test
Date	calendar dates	17/01/2021
Time	point in time	at night – at bedtime – at midday
Duration	length of time for occurrence	half an hour – twenty minutes
Set	frequency of occurrence	twice a week – every day

Table 3.1: Bete: Entities description and examples.

Documents	Sentences	Tokens	Entities	Relations
505	2340	55530	2396	1223

Table 3.2: Bete: Dataset information.

Entity	**Count**	**%**	**Average**	**Std**
Food	631	26.34	1.12	1.81
Complication	410	17.11	0.73	1.46
NonMedicalTreatment	405	16.90	0.72	1.05
Symptom	309	12.90	0.55	1.50
GlucoseValue	308	12.85	0.55	1.03
Time	71	2.96	0.13	0.47
Test	67	2.80	0.12	0.57
Medication	52	2.17	0.09	0.31
DiabetesType	48	2.00	0.09	0.58
Set	29	1.21	0.05	0.24
Insulin	25	1.04	0.04	0.32
Dose	23	0.96	0.04	0.24
Duration	18	0.75	0.03	0.21

Table 3.3: Bete Entities: Number of occurrences, percentage, average, and standard deviation annotation per document sorted in decreasing count order.

Relation	Count	%	Average	Std
has	833	68.11	1.48	3.19
treats	202	16.52	0.36	0.98
causes	79	6.46	0.14	0.74
diagnoses	52	4.25	0.09	0.46
prevents	50	4.09	0.09	0.53

Table 3.4: Bete Relations: Number of occurrences, percentage, average, and standard deviation annotation per document sorted in decreasing count order.

Entity	Description
Concept	identifies a relevant term, concept, idea, in the knowledge domain of the sentence.
Action	identifies a process or modification of other entities. It can be indicated by a verb or verbal construction, and also by nouns.
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 sentence or of different one.

Table 3.5: eHealth-KD: Entities description. Extracted from [31].

Relation	Description
is-a	indicates that one entity is a subtype, instance, or member of the class identified by the other.
same-as	indicates that two entities are semantically the same.
has-property	indicates that one entity has a given property or characteristic.
part-of	indicates that an entity is a constituent part of another.
causes	indicates that one entity provokes the existence or occurrence of another.
entails	indicates that the existence of one entity implies the existence or occurrence of another.
in-time	to indicate that something exists, occurs or is confined to a timeframe.
in-place	to indicate that something exists, occurs or is confined to a place or location.
in-context	to indicate a general context in which something happens, like a mode, manner, or state.
subject	indicates who performs the action.
target	indicates who receives the effect of the action.
domain	indicates the main entity on which the predicate applies.
arg	indicates an additional entity that specifies a value for the predicate to make sense.

Table 3.6: eHealth-KD: Relations description. Extracted from [31].

4
Models

A crucial part of developing any machine learning-related work is implementing models. Here we present our new Joint Entity and Relation Extraction model by providing details on input and output handling and architecture. Also, we explain the machine learning techniques that we applied and describe selected open-source models (e.g., from HuggingFace) that we use to benchmark the datasets.

4.1
Machine Learning Techniques

Here we explain some of the ML techniques used in this work. We opt to use classic methods, such as Conditional Random Fields and Support Vector Machines, and more recent ones, such as BERT Models.

4.1.1
Conditional Random Fields (CRF)

Conditional Random Fields is an ML sequence labeling model introduced by [17]. The model receives a sequence of elements as input: $x = (x_1, ..., x_M)$, and returns a label for each element: $y = (y_1, ..., y_M)$. Here we describe linear-chain CRF.

To build a CRF, we need feature functions, a function that takes in as input:

- the sequence of words x
- the position i of a word in the sequence
- the label y_i of the current word
- the label y_{i1} of the previous word

and outputs a real-valued number (though the numbers are often just either 0 or 1).

Next, we assign each feature function f_j a weight λ_j. and we can now score a labeling y of x by adding up the weighted features over all words in the sentence:

$$score(y|x) = \sum_{j=1}^{N} \sum_{i=1}^{M} \lambda_j f_j(x, i, y_i, y_{i-1})$$

where N is the number of features and M is the number of words in the sequence.

Finally, we can transform these scores into probabilities $p(y|x)$ between 0 and 1 by exponentiating and normalizing:

$$p(y|x) = \frac{exp[score(y|x)]}{\sum_{y'} exp[score(y'|x)]}$$

The features functions are detrimental components of CRF model and defining them depends on the task at hand. For example, considering the part-of-speech tag problem, feature functions could be:

- $f_1(x, i, y_i, y_{i-1}) = 1$ if y_{i-1}= Adjective and l_i= Noun; 0 otherwise.
- $f_2(x, i, y_i, y_{i-1}) = 1$ if $i = 1$, y_i= Verb and the sentence ends in a question mark; 0 otherwise.

Training a CRF is about learning the feature weights. The naive way to find optimal labeling is to calculate $p(y|x)$ for every possible labeling y, and then choose the label that maximizes this probability. For linear-chain CRFs, we can use a (polynomial-time) dynamic programming algorithm to find the optimal label, similar to the Viterbi algorithm for HMMs [14]. Also, we could use the gradient descent algorithm.

4.1.2 Support Vector Machine (SVM)

Support Vector Machine [15] is a supervised learning ML method that can work for classification or regression. For the classification task, which we will cover here, it defines the best decision boundary to separate data into classes.

More concretely, suppose we have a binary classification problem where $(x_1, y_1), (x_2, y_2), ..., (x_M, y_M)$ depicted data points with the feature vector $x_i \in R^D$ and the label $y_i \in \{-1, 1\}$. SVM tries to find the optimal hyperplane that correctly predicts y for unseen x.

Figure 4.1 shows an example where we can linearly separate data points into two classes; hence SVM successfully classifies each data point.

4.1.3 BERT Models

BERT, which stands for Bidirectional Encoder Representations from Transformers, is a language representation model [4]. Figure 4.2 shows the BERT architecture. The model generates deep bidirectional representations,

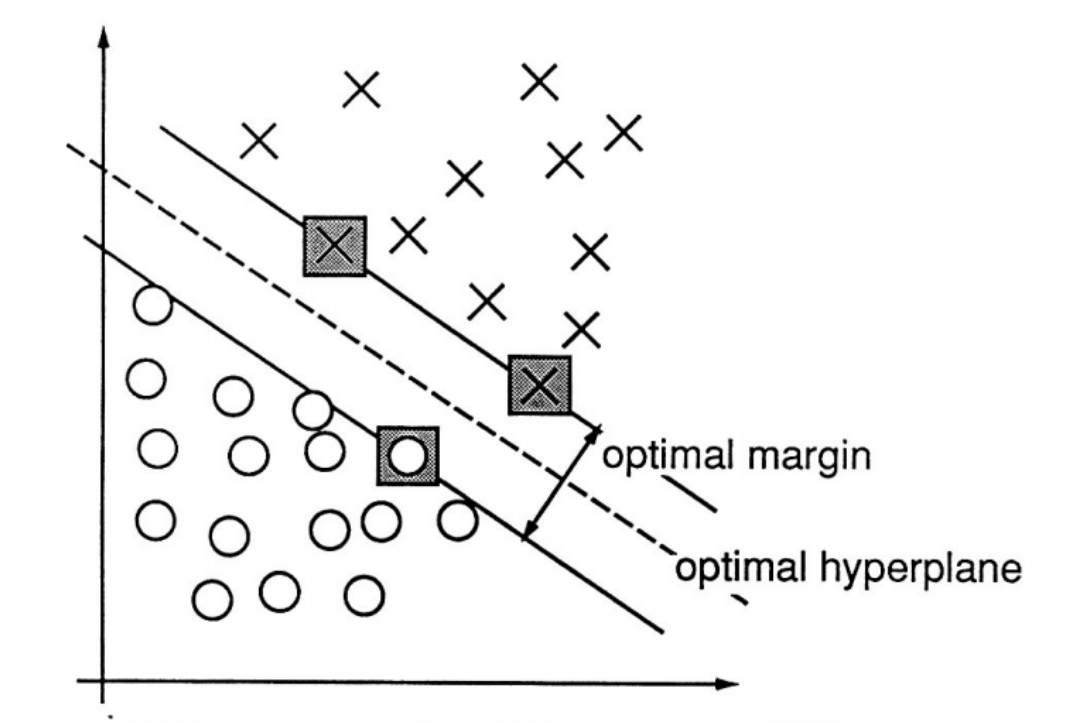

Figure 4.1: SVM classification example. Extracted from [51].

called embeddings, from an unlabeled text by jointly conditioning both left and right contexts in all layers. Once it is pre-trained, i.e., trained on a large quantity of data, it can be fine-tuned to the desired task, such as question answering, natural language inference, or named entity recognition.

BERT models use a subword segmentation algorithm to break a text into tokens, called WordPiece [50]. This algorithm consists of starting a vocabulary with only characters and iteratively adding the most frequent sequence of characters to the vocabulary. After tokenization, we can have subword tokens preceded by the symbol ##. Each pre-trained BERT model has its tokenizer. Figure 4.3 shows an example of mBERT tokenizer. We can see that doing the tokenization separates the words “encoded” and “pGKL” into subwords.

In this work, the BERT models are used to generate the embeddings to be utilized by downstream tasks. In our use case, we have three downstream tasks: Named Entity Recognition, Relation Extraction, and Joint Entity and Relation Extraction. We explain the specificities for each task in Chapter 4 .

We used a range of BERT models pre-trained in Portuguese, Spanish, or multilingual setup. Also, each is pre-trained on a large dataset from a specific or general domain. Below, we list each BERT-based model and provide a brief explanation:

– BERTimbau [1]: trained on brWAC [2], a large Portuguese corpus extracted from the Web.

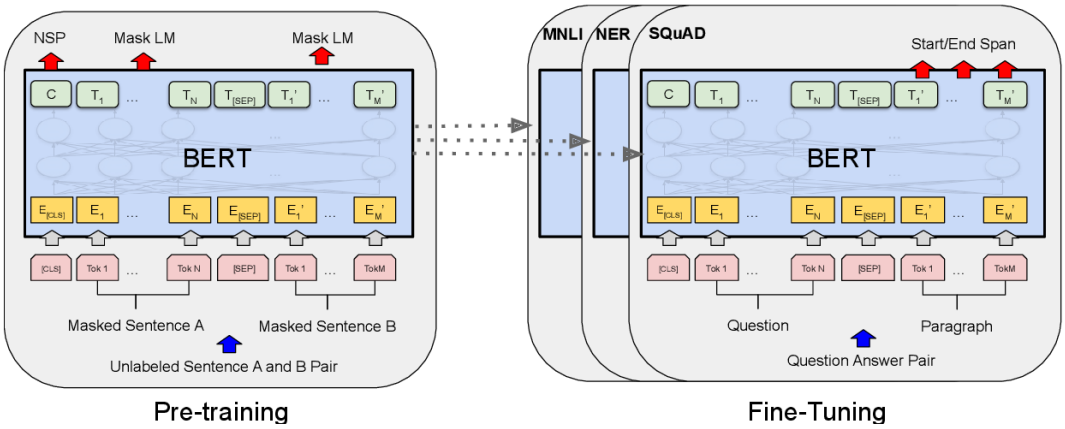

Figure 4.2: BERT architecture. Extracted from [4].

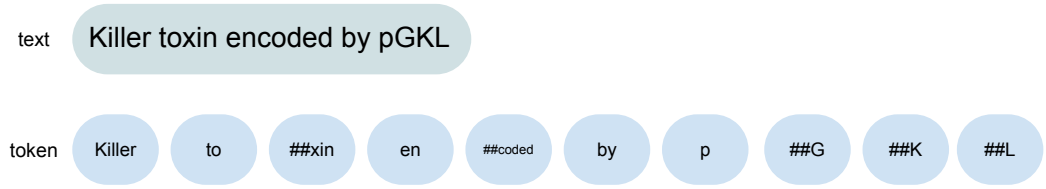

Figure 4.3: BERT tokenization example.

- BioBERTpt-bio [3]: trained on Portuguese biomedical texts.
- BioBERTpt-clin [3]: trained on clinical narratives from electronic health records from Brazilian Hospitals.
- BioBERTpt-all [3]: trained on both biomedical texts and clinical narratives.
- BETO [47]: trained on a big Spanish corpus [48].
- IXAmBERT [49]: trained on the English, Spanish and Basque Wikipedias, together with Basque crawled news articles from online newspapers.
- Multilingual BERT (mBERT) [4]: trained on the Wikipedia articles from 104 languages.

To use such models, we used the open-source HuggingFace [5] library, where we can access, fine-tune and evaluate the pre-trained models by writing Python code. In Table 4.1, we show the used models and their respective HuggingFace names.

Model name	HuggingFace name
BERTimbau	neuralmind/bert-base-portuguese-cased
BioBERTpt-bio	pucpr/biobertpt-bio
BioBERTpt-clin	pucpr/biobertpt-clin
BioBERTpt-all	pucpr/biobertpt-all
BETO	dccuchile/bert-base-spanish-wwm-cased
IXAmBERT	ixa-ehu/ixambert-base-cased
mBERT	bert-base-multilingual-cased

Table 4.1: Used BERT models.

4.2 Named Entity Recognition

Here we describe the models that address the NER task. Usually, there are two steps that NER models must perform: tokenize the source text and assign named entities to each token.

4.2.1 CRF

Condition Random Fields (CRF) is a model suited for NER task. First, we split the sentence into a list of tokens. We used a Python library called Spacy [43] with models for Portuguese, Spanish, and English.

Then, we extract a set of features; the model can classify each token as one of the entities or none. Table 4.2 shows the used set of features.

Feature
Part of speech tagging, extracted using Spacy [43]. Table 4.3 shows the used classes.
If the word is in uppercase.
If the word is a digit.
If the word is a title, i.e. start with an uppercase letter and the rest is in lowercase.
The previous word and the above features.
The next word and the above features.

Table 4.2: Used CRF features.

4.2.2 BERT for Named Entity Recognition

Each of the BERT models described in 4.1.3 creates an embedding from a sentence. Therefore, we need a classification head on top of the BERT network to address the NER task.

Tag	Description
ADJ	adjective
ADV	adverb
INTJ	interjection
NOUN	noun
PROPN	proper noun
VERB	verb
PRON	pronoun
SCONJ	subordinating conjunction
ADP	preposition/postposition
AUX	auxiliary
CONJ	coordinating conjunction
DET	determiner
NUM	numeral
PART	particle
PUNCT	punctuation
SYM	symbol
X	unspecified POS

Table 4.3: Used Part of speech tagging. Extracted from [44].

Figure 4.4 shows the architecture of BERT for the NER model. The classification head, which is between the tags and encoder representation in the figure, is a dense layer that takes the embeddings and returns the probabilities of the tags. So after the classification head, we need a softmax layer to get the predicted tag.

4.3 Relation Extraction

To build a relation extraction model, we must first parse the data as input to the models. For this, we need the output from the named entity recognition model. Also, we need the annotated data that assigns a relationship to a pair of entities. With that, we conduct the following preprocessing steps:

Identify the source and target entities using an out-of-vocabulary word for each annotated relation. For example, replacing with ENTITY1 and ENTITY2 the source and target entity, respectively. Also, another approach is to mark the entities using tags such as <e1>“source entity”</e1> and <e2>“target entity”</e2>.

For example, consider the sentence: “The killer virus is a satellite RNA of L-A and is totally dependent on 231 L-A proteins for replication”, the entities “virus” and “satellite RNA”, and the relationship “is-a” between the two entities. After preprocessing, we would have: “The killer [E1]virus[/E1] is

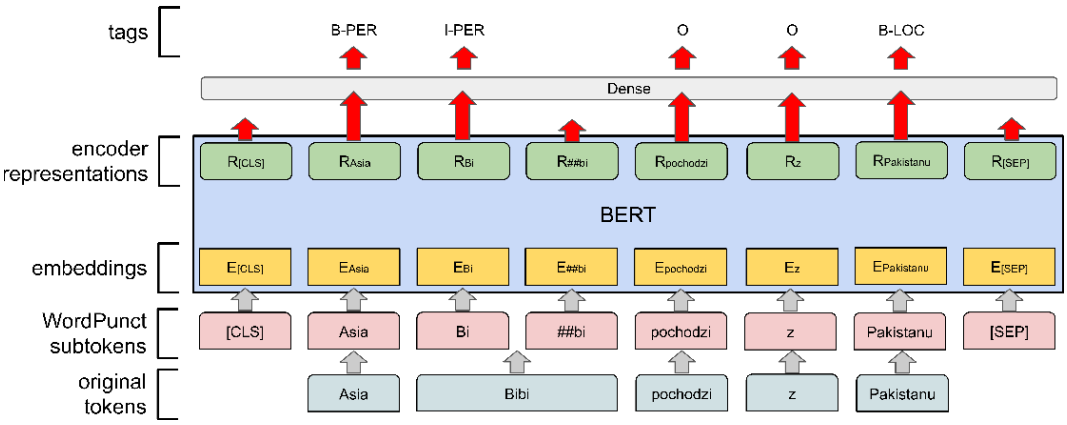

Figure 4.4: BERT For Named Entity Recognition architecture. Extracted from [45].

a [E2]satellite RNA[/E2] of L-A and is totally dependent on 231 L-A proteins for replication"

To train the relation extraction model, we also need to add negative examples, i.e., examples where we have two entities and no relation between them. By performing this, our model can decide to classify as one of the available relation labels or none, identified by the "O" tag. The number of negative examples to add varies; it affects if the model becomes biased toward the relation labels or the "O" tag. We performed different experiments with this number and report in Section 6.2.

4.3.1 SVM

We must extract features from the text to use an SVM model for Relation Extraction. These features can be defined in many ways. For example, features can be lexical, semantic, or syntactic, such as if the word is lowercased or the part-of-speech of the word. In our use case, we opt to use bag-of-words as feature.

So after the preprocessing steps, we extract bag-of-words features from the text and feed them to the SVM model.

4.3.2 BERT for Relation Extraction

The BERT for Relation Extraction is a model proposed by [32]. The model learns relation representations directly from the text by applying a method of training called matching the blanks.

By considering sentences with tagged entities, the method creates training data by replacing the text between entities' tags with a unique [BLANK]

symbol. For example, considering the input sentence: “The killer [E1]virus[/E1] is a [E2]satellite RNA[/E2] of L-A and is totally dependent on 231 L-A proteins for replication”, after applying this step the sentence would be: “The killer [BLANK] is a [BLANK] of L-A and is totally dependent on 231 L-A proteins for replication”.

Then, the data is fed to a BERT-based model, and the network tries to learn the representation of each relation type.

4.4 Joint Entity and Relation Extraction model

The Joint model consists of a multilingual BERT-based system that jointly predicts entities and relations. The model works end-to-end: it learns the steps to transform text input into entity and relationship tags.

During training, the proposed system is fine-tuned in 3 sequential steps: the first prioritizes the entity recognition task, and the second prioritizes relation extraction. Finally, the last step trains both tasks using a multi-task strategy.

This section details input handling, system architecture, and output handling and presents the parameters and training setup.

4.4.1 Input handling

To train the model, we receive as input a sentence, a list of entities with the character span-based information where the entity starts and ends, and a list of relations where each relation spans two entities.

Since our network works at token level, first, we tokenize the sentence text using BERT default tokenizer [4], resulting in WordPiece [50] information.

Next, for each token, we assign Begin and Inside tags (IOB2 format), if it is part of an entity, and `O` otherwise. Using this approach, we can represent consecutive entities with more than one token. However, this prevents us from representing discontinuous entities e.g., considering the text span “uno o dos días”, we cannot represent the entity (“uno días”) using the IOB2 format. Therefore, we only consider the first entity (“uno”). We opt for this simple approach instead of a complex representation because we value building a more efficient and straightforward model.

Also, for subwords tokens, we consider only the first one as part of the entity and the rest as “O”. For example, for the tokens: g, ##lut, and ##en, only the token “g” is annotated as the Concept entity; “##lut” and “##en” have O value.

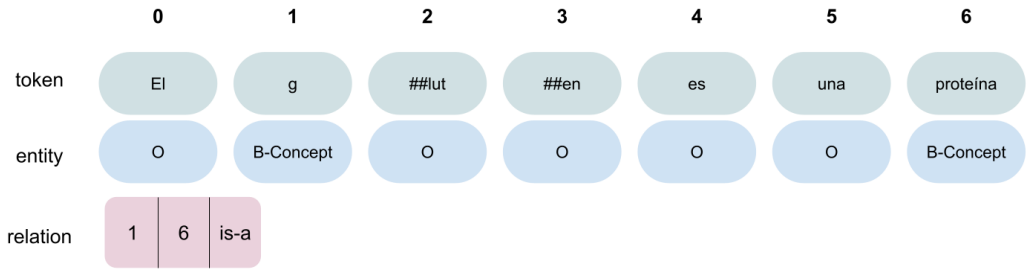

Figure 4.5: Example of model's input for the sentence: "El gluten es una proteína".

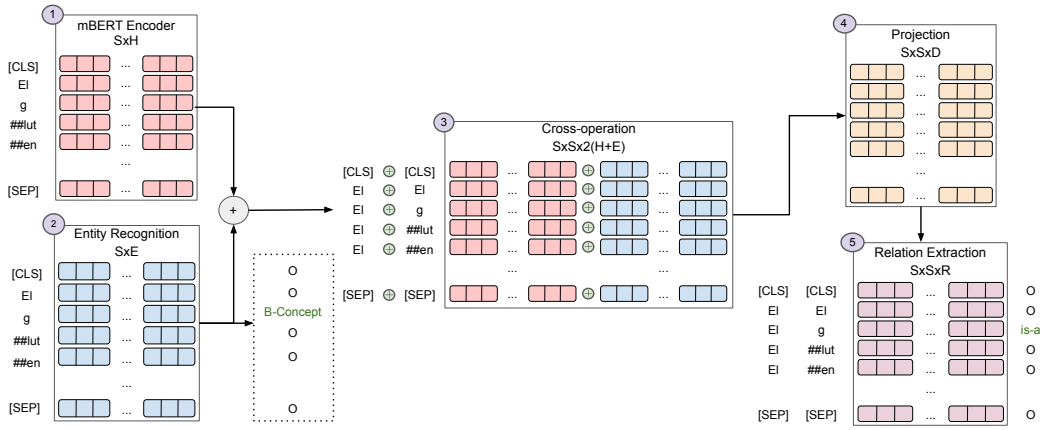

Figure 4.6: Architecture of the novel joint entity and relation extraction model.

As for relations, we represent them as triples, containing the first token of each entity in the relationship and the relation type. We use these triples to fill a relation matrix that we describe next. Figure 4.5 shows an example input of our model.

4.4.2 Architecture

The architecture of our model is presented in Figure 4.6. In the following paragraphs, we explain each of its components:

Encoder Our approach first tokenizes the input text and encodes its tokens (words or subwords) into vector representations (Step 1 in Figure 4.6). After conducting several experiments testing different BERT models, we opt to use mBERT, a multilingual version of BERT pretrained in texts of 104 languages [4]. We used the `bert-base-multilingual-cased` setting, with 12 self-attention heads, 12 layers (transformer blocks), and an embedding length of 768, which encodes multilingual cased texts. After performing this step, we get a SxH, where S is the number of tokens in the sentence and H is the

embedding length.

Entity Recognition Once the input text is encoded, as depicted in Step 2 in Figure 4.6, the encoded vector representations are fed into the entity recognition classifier. For example, considering the eHealth-KD dataset, each entity can be classified into 4 categories: `Concept`, `Action`, `Predicate` and `Reference`. As was mentioned on Section 4.4.1, we used the `IOB2` format, popular in Named Entity Recognition applications, to label each token according to 9 categories: `O`, `B-Concept`, `I-Concept`, `B-Action`, `I-Action`, `B-Predicate`, `I-Predicate`, `B-Reference` and `I-Reference`. The `O` label is used to mark tokens which are not part of an entity mention, whereas the ones starting with `B-` and `I-` indicate the beginning and subsequent tokens of a mention, respectively.

Relation Extraction We concatenate the logits, i.e. predictions before applying softmax layer, of the entity recognition classifier with the vector representations related to the respective tokens. A cross-operation is then performed by concatenating each pair of vector representations among the tokens, resulting in a tensor of dimension $SxSx2(H + E)$, being S the number of tokens in the sentence, H the 768 dimensions of the vector representations and E the 9 dimensions of the logits (Step 3 of Figure 4.6). The matrix is further fed into a projection layer with an `Tanh` activation function, which maps the input onto a $SxSxD$, where $D = 768$ (Step 4 of Figure 4.6). Finally, the output of the previous operation is given as input to the relation extraction classifier which predicts the relation of each pair of tokens, for the eHealth-KD dataset, according to 13 categories (`is-a`, `part-of`, `has-property`, `causes`, `entails`, `in-context`, `in-place`, `in-time`, `subject`, `target`, `domain`, `arg` and `same-as`), plus a `O` one, which indicates there is no relation among the target pair.

Classifiers Both entity recognition and relation extraction classifiers consist in a projection layer with a `Mish` [33] activation function and dropout of 0.2, followed by a softmax layer.

4.4.3 Output handling

The model's output must be converted back to a character span-based format. So we implement a postprocessing module that is responsible for this conversion. The model's output contains a sequence of tokens, each assigned

to an entity tag and a SxS matrix informing the relationship between each pair of tokens.

For each token, if it is the beginning of an entity, we identify the character range it spans and add this span list to the result. Next, we discard entities that are entirely contained within another one and which start with a stopword. Lastly, we construct the relations by linking entities that contain at least one token in the model's relations output.

4.4.4 Parameters and Training Setup

Our neural network approach was trained using the `AdamW` [34] optimizer combined with a linear scheduler which warms up the training process from an initial learning rate of 2e-6 up to 2e-5 over the first 10 epochs. Using a batch of size 1, we train the approach in 3 sequential steps.

In the first step, all the training parameters of the network are frozen except for the ones from mBERT and the entity recognition classifier. The model is then trained for 50 epochs with early stopping of patience 15 (i.e., the training algorithm waits for 15 epochs before early stop if there is no progress on the validation set), computing the loss only based on the entity recognition task:

$$J_{ent}(x^{(ent)}, y^{(ent)}) = \frac{1}{N}\sum_{n=1}^{N} x^{(ent)}_{y_n^{(ent)}} \tag{4-1}$$

where $x^{(ent)}$ is the likelihood computed by the entity recognition classifier, $y^{(ent)}$ is the gold standards and N is the size of the batch. The notation $x^{(ent)}_{y_n^{(ent)}}$ means that we are analyzing the classifier likelihood for each gold standard y_n in the batch.

For the second step, which focuses on the relation extraction task, we only freeze the training parameters of the entity recognition classifier. The approach is also trained for 50 epochs with early stopping of patience 15, though, unlike the previous step, the loss is computed based on the relation extraction task:

$$J_{rel}(x^{(rel)}, y^{(rel)}) = \frac{1}{N}\sum_{n=1}^{N} x^{(rel)}_{y_n^{(rel)}} \tag{4-2}$$

where $x^{(rel)}$ is the likelihood computed by the relation extraction classifier, $y^{(rel)}$ the gold standards, and N is the size of the batch.

Finally, we perform a third training step with 100 epochs and early stopping of patience 15 to fine-tune the model for both tasks. None of the training parameters are frozen and the loss is computed based on [46] in the following way:

$$J = e^{-\alpha_{ent}} \times J_{ent}(x^{(ent)}, y^{(ent)}) + \alpha_{ent} + e^{-\alpha_{rel}} \times J_{rel}(x^{(rel)}, y^{(rel)}) + \alpha_{rel} \quad (4\text{-}3)$$

being α_{ent} and α_{rel} training parameters as well.

5
Experimental Setup

Here we describe our setup for all experiments. First, we explain the data training, development, and testing split. Then training setup from the baselines and BERT models. Finally, we explain the evaluation setup and metrics.

5.1
Data

We evaluated models on two datasets, Bete and eHealth-KD. Considering Bete, we randomly divided the 505 documents into train/dev/test using the split 0.8/0.1/0.1, respectively. For eHealth-KD, we used 1500 sentences in the training, 100 in the development, and 300 in the testing set.

We tuned the hyperparameters on the development set and reported the results on the test set.

5.2
Training and Evaluation setup

First, regarding the baseline models, we used a CRF model with 0.1 as the coefficient of L1 and L2 regularization, training with gradient descent using the L-BFGS method for a maximum of 100 iterations.

We employed a radial basis function (RBF) kernel for the SVM model with a one-vs-rest decision function. As features, we extracted TF-IDF features considering only 3-gram words.

For BERT models for the NER task, we used the Adam [34] optimizer with a learning rate of 1e-5 and a maximum length of 512. Moreover, we trained for 50 epochs with early stopping of 15 epochs and a batch size of 64.

Considering BERT for Relation Extraction, we used 7e-5 as the learning rate with Adam [34] optimizer and max length of 512, training for 11 epochs and a batch size of 32.

We considered the following metrics for evaluation: precision, recall, and F1 score. We reported the weighted average F1 score, averaging the support-weighted mean per label. In addition, we computed results considering all classes and, for the best model, we measured each label independently.

6
Results and Discussion

In this section, we show our experiments and results. First, we divide this chapter by task (NER, RE, and Joint Entity and Relation Extraction), and then we show the results for both datasets (Bete and eHealth-KD).

6.1
Named Entity Recognition

We conducted experiments using methods to recognize entities on the Bete and eHealth-KD data. For both, we evaluated a baseline CRF model. Considering Bete data, since the dataset is in Brazilian Portuguese, we chose deep learning models trained on multilingual and Brazilian Portuguese data. These models are multilingual BERT (mBERT), BERTimbau, and the three different versions of BioBERTpt: BioBERTpt-bio, trained on Portuguese biomedical texts, BioBERTpt-clin, trained on clinical narratives from electronic health records from Brazilian Hospitals, and BioBERTpt-all, trained on both biomedical texts and clinical narratives.

The data in eHealth-KD is in Spanish and English, so we used multilingual and Spanish BERT versions: mBERT and BETO, trained on a Spanish corpus from various domains: Wikipedia, news, and subtitles.

Table 6.1 shows the results for Bete dataset. The best F1-score is the BioBERTpt-clin model, outperforming BioBERTpt-all by 0.9 points. Also, BERTimbau did not perform well, with an F1-score 5.3 points lower than the second-lowest.

We also analyzed the metrics per entity type for the best performing model, as it is shown in Table 6.2. The model produced good results (>80%) for the three entities with the most examples: Food, Complication, and Symptom. However, for entities with few examples, the model did not predict well (<60%): Test, Time, and Set.

Considering the eHealth-KD dataset, table 6.3 shows the results. We can see that model trained only on Spanish text (BETO) did not perform well, which can be explained by the data containing Spanish and English sentences. IXAmBERT and mBERT are almost identical.

Model	**Precision (%)**	**Recall (%)**	**F1 (%)**
CRF	**80.3**	72.9	76.1
BioBERTpt-bio	73.1	80.5	76.6
BioBERTpt-clin	77.5	81.8	**79.4**
BioBERTpt-all	74.5	**83.2**	78.5
BERTimbau	72.2	70.2	70.8
mBERT	74.3	81.6	77.6

Table 6.1: Bete: Experiments for all models. Best scores are highlighted in bold.

Entities	**Precision (%)**	**Recall (%)**	**F1 (%)**	**N° of examples**
Complication	84.8	91.8	88.2	61
DiabetesType	100.0	100.0	100.0	2
Dose	50.0	100.0	66.7	2
Duration	100.0	100.0	100.0	1
Food	79.1	81.5	80.3	135
GlucoseValue	60.4	78.4	68.2	37
Insulin	100.0	100.0	100.0	4
Medication	68.8	91.67	78.6	12
NonMedicalTreatment	72.7	64.9	68.6	37
Set	66.7	50.0	57.1	4
Symptom	90.0	94.7	92.3	57
Test	50.0	33.3	40.0	9
Time	44.4	50.0	47.1	8
Weighted Average	77.5	81.8	79.4	369

Table 6.2: Bete: Detailed results for the best model (BioBERTpt-clin).

Model	**Precision (%)**	**Recall (%)**	**F1 (%)**
CRF	48.4	45.5	46.7
BETO	53.2	70.2	60.4
IXAmBERT	**63.9**	70.6	66.8
mBERT	61.8	**73.1**	**66.9**

Table 6.3: eHealth-KD: Experiments for all models. Best scores are highlighted in bold.

Table 6.4 shows the results for the best performing model (mBERT). The result for entities with most examples (Concept and Action) is better than those with few examples (Predicate and Reference). This reflects the need for different entities' data to improve the model's performance.

We can notice that the CRF model has the best precision score in Bete

but falls short in eHealth-KD, having the worst performance. One explanation is that Bete data is much smaller than eHealth-KD, which makes simpler models more precise.

Entities	Precision (%)	Recall (%)	F1 (%)	N° of examples
Action	55.0	76.6	64.0	449
Concept	65.5	75.2	70.1	1938
Predicate	45.0	47.7	46.3	199
Reference	19.4	36.8	25.5	19
Weighted Average	61.8	73.1	66.9	2605

Table 6.4: eHealth-KD: Detailed results for the best model (mBERT).

6.2 Relation Extraction

To evaluate the relation extraction task, we experimented using the two models mentioned in 4.3: SVM and BERT for Relation Extraction (BERT-RE), using mBERT as the BERT encoder.

Table 6.5 shows the result for both models on Bete data and table 6.6 on eHealth-KD. We can observe that, although SVM has higher precision than BERT-RE, the BERT-RE model performs better overall. The structure using BERT embeddings helps the model encode the input and understand which relation is present.

Tables 6.7 and 6.8 describe an in-depth analysis of BERT-RE performance for each relation type.

Figures 6.1 and 6.2 show the result per epoch using BERT-RE model. The performance increases with the number of epochs; for Bete, the results are more stable than eHealth-KD.

Model	Precision (%)	Recall (%)	F1 (%)
SVM	**80.0**	46.2	58.1
BERT-RE (mBERT)	73.6	**77.8**	**75.1**

Table 6.5: Bete: Experiments for all models. Best scores are highlighted in bold.

Model	Precision (%)	Recall (%)	F1 (%)
SVM	27.6	4.4	6.4
BERT-RE (mBERT)	**53.3**	**50.8**	**48.6**

Table 6.6: eHealth-KD: Experiments for all models. Best scores are highlighted in bold.

Entities	Precision (%)	Recall (%)	F1 (%)	N° of examples
causes	33.33	33.33	33.33	6
prevents	66.67	50.00	57.14	4
treats	52.94	6429	58.06	14
has	77.89	86.05	81.77	86
diagnoses	100.00	57.14	72.73	7
Weighted Average	73.56	77.78	75.06	117

Table 6.7: Bete: Detailed results for the best model (BERT-RE).

6.3 Joint Entity and Relation Extraction

We evaluated the Joint model on the eHealth-KD dataset and compared it with other participants' systems from the IberLEF eHealth Knowledge Discovery Challenge 2021 [31].

The challenge was divided into three tasks:

- Task A: Entity Recognition.
- Task B: Relation Extraction.
- Main: Evaluates tasks A and B together as a pipeline.

Table 6.9 displays the results (precision, recall, and F1) reported by the participating systems in the eHealth challenge 2021. For the entity recognition task (Task A), our approach, which ran for 67 epochs in the third training step and is labeled as `Our Approach` in the table, ranked first with an F1 of 70.60% outperforming `Vicomtech`, the second best in the task and developed by the winning team of the 2020 version of the challenge [31]. On the other hand, for the relation extraction task (Task B) our system had a significant drop, ranking 4^{th} in the task with an F1 of 26.32% behind the `EdgardAndres (IXA)`, `Vicomtech` and `uhKD4` systems. The intermediate performance of our approach in task B was made up for by its good performance for the entity recognition task so that our approach ranked second in the Main task (which combines both tasks), just behind `Vicomtech`.

Entities	Precision (%)	Recall (%)	F1 (%)	N° of examples
is-a	46.39	67.16	54.88	67
part-of	100.00	12.50	22.22	24
has-property	63.16	14.63	23.76	82
causes	76.47	48.15	59.09	27
entails	25.00	21.43	23.08	14
in-context	47.44	37.37	41.81	198
in-place	41.18	44.44	42.75	63
in-time	48.84	84.00	61.76	25
subject	60.53	66.99	63.59	103
target	54.55	66.67	60.00	162
domain	52.94	72.97	61.36	37
arg	23.81	60.00	34.09	25
same-as	66.67	72.73	69.57	11
Weighted Average	53.30	50.84	48.63	838

Table 6.8: eHealth-KD: Detailed results for the best model (BERT-RE).

	Main				**Task A**				**Task B**			
	Entity + Relation				**Entity R.**				**Relation E.**			
	#R	**P**	**R**	**F1**	**#R**	**P**	**R**	**F1**	**#R**	**P**	**R**	**F1**
Our Approach	2	**56.85**	50.28	52.84	1	**71.49**	69.73	**70.60**	4	36.66	20.54	26.32
Our Approach 100 epcs	-	53.63	49.39	51.42	-	**71.49**	69.20	70.33	-	32.31	22.96	26.85
Vicomtech [35]	1	54.08	53.46	**53.11**	2	69.99	**74.71**	68.41	2	54.19	28.31	37.19
EdgarAndres (IXA) [36]	3	46.46	**53.86**	49.89	3	61.37	69.8	65.33	1	45.36	**40.95**	**43.04**
uhKD4 [37]	4	48.53	37.43	42.26	5	51.75	53.74	52.73	3	**55.62**	22.24	31.77
UH-MMM [38]	5	29.16	40.37	33.87	4	54.60	68.50	60.77	5	07.73	4.13	05.38
CodestrangeTeam [39]	6	33.70	17.69	23.20	10	41.50	4.44	8.02	6	43.75	1.70	3.28
baseline [31]	7	33.70	17.69	23.20	7	35.03	27.17	30.60	7	43.75	1.70	3.28
JAD [40]	8	10.95	23.44	7.14	8	31.58	22.46	26.25	8	37.50	0.365	0.722
GuanZhengyi [41]	-	-	-	-	6	52.04	24.60	33.41	-	-	-	-
Maoqin [42]	-	-	-	-	9	27.11	12.73	17.32	-	-	-	-

Table 6.9: Participating systems' reported results for the eHealth-KD Challenge 2021.

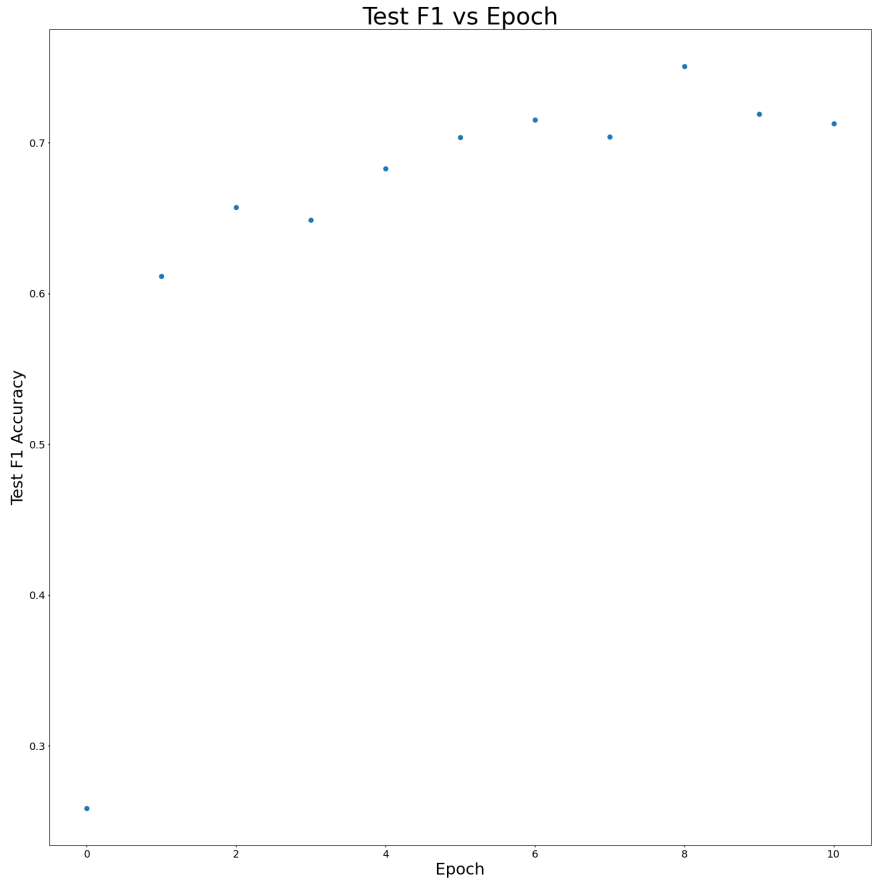

Figure 6.1: Bete: F1-score per epoch using BERT-RE model.

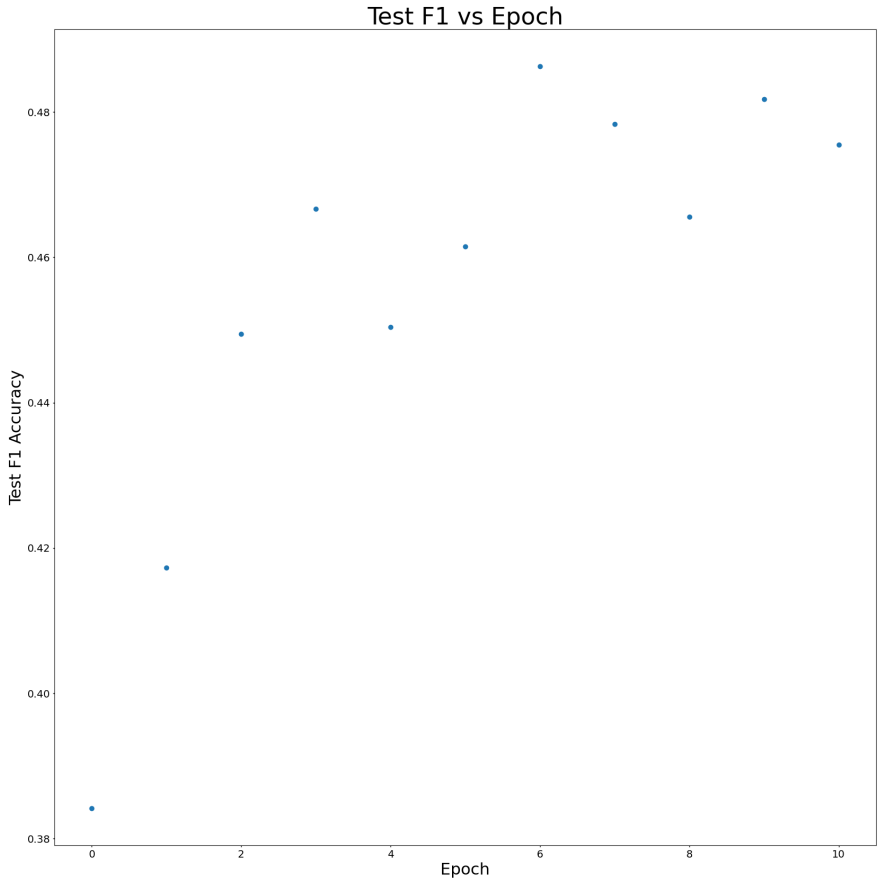

Figure 6.2: eHealth-KD: F1-score per epoch using BERT-RE.

7
Conclusions

In this thesis, we worked on Named Entity Recognition, Relation Extraction, and Joint Entity and Relation Extraction tasks. These are essential for the Information Extraction research community, as they are the basis of other downstream tasks such as recommender systems and question answering.

We focused on creating a novel Portuguese dataset in the biomedical domain, Bete. The corpus was annotated by undergraduate students from different courses, i.e., linguistics, and medicine, for NER and RE tasks. We measured statistics and performed the first evaluation using up-to-date NER and RE models based on the Transformer architecture.

Also, we developed a novel model for the Joint Entity and Relation Extraction task, called the Joint model. We used state-of-the-art techniques such as multi-task training and generalized the method to work in many languages using multilingual BERT embeddings. We published a research paper at IberLEF Challenge 2021 [31], winning first place considering only the NER task and second place overall.

For the results, we showed that neural-based models perform better on the NER and RE tasks when compared to machine learning methods. Also, we showed that the domain of the pre-trained data impacts the result of the model; for example, a model pre-trained on biomedical text performs better than a model pre-trained on news text when evaluating data from the biomedical domain.

This thesis opens the path to more works using non-English languages for NER and RE tasks, such as Portuguese and Spanish. Also, other evaluations would be valuable, for example, by using a metric to measure ethical values, like gender or race.

Bibliography

[1] SOUZA, F.; NOGUEIRA, R. ; LOTUFO, R.. **BERTimbau: Pretrained BERT Models for Brazilian Portuguese**. In: Cerri, R.; Prati, R. C., editors, INTELLIGENT SYSTEMS, p. 403–417, Cham, 2020. Springer International Publishing.

[2] WAGNER FILHO, J. A.; WILKENS, R.; IDIART, M. ; VILLAVICENCIO, A.. **The BRWAC corpus: A new open resource for Brazilian Portuguese**. LREC 2018 - 11th International Conference on Language Resources and Evaluation, (May):4339–4344, 2019.

[3] SCHNEIDER, E. T. R.; DE SOUZA, J. V. A.; KNAFOU, J.; OLIVEIRA, L. E. S. E.; COPARA, J.; GUMIEL, Y. B.; OLIVEIRA, L. F. A. D.; PARAISO, E. C.; TEODORO, D. ; BARRA, C. M. C. M.. **BioBERTpt - A Portuguese Neural Language Model for Clinical Named Entity Recognition**. p. 65–72, 2020.

[4] DEVLIN, J.; CHANG, M. W.; LEE, K. ; TOUTANOVA, K.. **BERT: Pre-training of deep bidirectional transformers for language understanding**. NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference, 1(Mlm):4171–4186, 2019.

[5] WOLF, T.; DEBUT, L.; SANH, V.; CHAUMOND, J.; DELANGUE, C.; MOI, A.; CISTAC, P.; RAULT, T.; FUNTOWICZ, M.; DAVISON, J.; SHLEIFER, S.; PLATEN, P. V.; MA, C.; JERNITE, Y.; PLU, J.; XU, C.; SCAO, T. L.; GUGGER, S.; DRAME, M.; LHOEST, Q. ; RUSH, A. M.. **Transformers : State-of-the-Art Natural Language Processing**. p. 38–45, Oct. 2020.

[6] FUKUDA, K.; TAMURA, A.; TSUNODA, T. ; TAKAGI, T.. **Toward information extraction: identifying protein names from biological papers.** Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing, p. 707–718, 1998.

[7] GAIZAUSKAS, R.; DEMETRIOU, G.; ARTYMIUK, P. J. ; WILLETT, P.. **Protien structures and information extraction from biological texts: The PASTA system**. Bioinformatics, 19(1):135–143, 2003.

[8] HANISCH, D.; FUNDEL, K.; MEVISSEN, H. T.; ZIMMER, R. ; FLUCK, J.. **ProMiner: Rule-based protein and gene entity recognition**. BMC Bioinformatics, 6(SUPPL.1):1–9, 2005.

[9] FARMAKIOTOU, D.; KARKALETSIS, V.; KOUTSIAS, J.; SIGLETOS, G.; SPYROPOULOS, C. D. ; STAMATOPOULOS, P.. **Rule-based named entity recognition for Greek financial texts**. Proc. of the Workshop on Computational lexicography and Multimedia Dictionaries (COMLEX 2000), p. 75–78, 2000.

[10] SCIENTIFIC, T.; YAZICI, A.. **Rule-based Named Entity Recognition from Turkish Texts**. (January 2009), 2015.

[11] RIAZ, K.. **Rule-Based Named Entity Recognition in Urdu**. Proceedings of the 2010 Named Entities Workshop, (July):126–135, 2010.

[12] EFTIMOV, T.; SELJAK, B. K. ; KOROŠEC, P.. **A rule-based named-entity recognition method for knowledge extraction of evidence-based dietary recommendations**, volumen 12. 2017.

[13] POPOVSKI, G.; KOCHEV, S.; SELJAK, B. K. ; EFTIMOV, T.. **Foodie: A rule-based named-entity recognition method for food information extraction**. ICPRAM 2019 - Proceedings of the 8th International Conference on Pattern Recognition Applications and Methods, (Icpram):915–922, 2019.

[14] EDDY, S. R.. **Hidden Markov models**. Current Opinion in Structural Biology, 6(3):361–365, 6 1996.

[15] PLATT;, B. S. . S. T. D. E. O. J. M. A. H.. **Support Vector Machines**. Annual review of psychology, 13:107–144, 1998.

[16] QUINLAN, J. R.. **Induction of decision trees**. Machine Learning 1986 1:1, 1(1):81–106, 3 1986.

[17] LAFFERTY, J.; MCCALLUM, A. ; PEREIRA, F.. **Conditional Random Fields : Probabilistic Models for Segmenting and Labeling Sequence Data**. 2001(June):282–289, 1999.

[18] ZHOU, G.; SU, J.. **Named entity recognition using an HMM-based chunk tagger**. (July):473, 2001.

[19] ISOZAKI, H.; KAZAWA, H.. **Efficient support vector classifiers for named entity recognition**. p. 1–7, 2002.

[20] CARRERAS, X.; MÀRQUEZ, L. ; PADRÓ, L.. **Named Entity Extraction using AdaBoost**. p. 1–4, 2002.

[21] SCHAPIRE, R. E.; SINGER, Y.. **Improved boosting algorithms using confidence-rated predictions**. Machine Learning, 37(3):297–336, 1999.

[22] KRISHNAN, V.; MANNING, C. D.. **An effective two-stage model for exploiting non-local dependencies in named entity recognition**. COLING/ACL 2006 - 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, 1(July):1121–1128, 2006.

[23] KAMBHATLA, N.. **Combining Lexical, Syntactic, and Semantic Features with Maximum Entropy Models for Extracting Relations**. Proceedings of the ACL 2004, p. 22–es, 2004.

[24] ZHOU, G. D.; SU, J.; ZHANG, J. ; ZHANG, M.. **Exploring various knowledge in relation extraction**. ACL-05 - 43rd Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, (June):427–434, 2005.

[25] JIANG, J.; ZHAI, C. X.. **A systematic exploration of the feature space for relation extraction**. NAACL HLT 2007 - Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics, Proceedings of the Main Conference, (April):113–120, 2007.

[26] ZELENKO, D.; AONE, C. ; RICHARDELLA, A.. **Kernel Methods for Relation Extraction**. Journal of Machine Learning Research, 3(6):1083–1106, 2003.

[27] BUNESCU, R. C.; MOONEY, R. J.. **Subsequence kernels for relation extraction**. Advances in Neural Information Processing Systems, p. 171–178, 2005.

[28] ECKART DE CASTILHO, R.; MÙJDRICZA-MAYDT, ; YIMAM, S. M.; HARTMANN, S.; GUREVYCH, I.; FRANK, A. ; BIEMANN, C.. **A Web-based Tool for the Integrated Annotation of Semantic and Syntactic Structures**. Proceedings of the workshop on Language Technology Resources and Tools for Digital Humanities (LT4DH) at COLING 2016, p. 76–84, 2016.

[29] BEN ABACHA, A.; ZWEIGENBAUM, P.. **MEANS: A medical question-answering system combining NLP techniques and semantic Web technologies**. Information Processing and Management, 51(5):570–594, 2015.

[30] STENETORP, P.; PYYSALO, S. ; TOPI, G.. **BRAT : a Web-based Tool for NLP-Assisted Text Annotation**. (Figure 1):102–107, 2012.

[31] PIAD-MORFFIS, A.; GUTIÉRREZ, Y.; CAÑIZARES-DIAZ, H.; ESTÉVEZ-VELARDE, S.; MONTOYO, A. ; ALMEIDA-CRUZ, Y.. **Overview of the ehealth knowledge discovery challenge at iberlef 2020**. CEUR Workshop Proceedings, 2664:85–93, 2020.

[32] SOARES, L. B.; FITZGERALD, N.; LING, J. ; KWIATKOWSKI, T.. **Matching the blanks: Distributional similarity for relation learning**. ACL 2019 - 57th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, p. 2895–2905, 2020.

[33] MISRA, D.. **Mish: A Self Regularized Non-Monotonic Activation Function**. 2019.

[34] KINGMA, D. P.; BA, J. L.. **Adam: A method for stochastic optimization**. 3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings, p. 1–15, 2015.

[35] GARCÍA-PABLOS, A.; PÉREZ, N. ; CUADROS, M.. **Vicomtech at ehealth-kd challenge 2021: Deep learning approaches to model health-related text in spanish**. In: PROCEEDINGS OF THE IBERIAN LANGUAGES EVALUATION FORUM (IBERLEF 2021), 2021.

[36] ANDRÉS, E.. **Ixa at ehealth-kd challenge 2021: Generic sequence labeling as relation extraction approach**. In: PROCEEDINGS OF THE IBERIAN LANGUAGES EVALUATION FORUM (IBERLEF 2021), 2021.

[37] ALFARO-GONZÁLEZ, D.; PÉREZ-PERERA, D.; GONZÁLEZ-RODRÍGUEZ, G. ; OTAÑO-BARRERA, A. J.. **uhkd4 at ehealth-kd challenge 2021: Deep learning approaches for knowledge discovery from spanish biomedical documents**. In: PROCEEDINGS OF THE IBERIAN LANGUAGES EVALUATION FORUM (IBERLEF 2021), 2021.

[38] MONTEAGUDO-GARCÍA, L.; MARRERO-SANTOS, A.; FERNÁNDEZ-ARIAS, M. S. ; CAÑIZARES-DÍAZ, H.. **Uh-mmm at ehealth-kd challenge 2021**. In: PROCEEDINGS OF THE IBERIAN LANGUAGES EVALUATION FORUM (IBERLEF 2021), 2021.

[39] MARTI, R.; BERMUDEZ, C.; GARCÍA, L. ; GUTIÉRREZ, L.. **Codestrange at ehealth-kd challenge 2021**. In: PROCEEDINGS OF THE IBERIAN LANGUAGES EVALUATION FORUM (IBERLEF 2021), 2021.

[40] NAVARRO COMABELLA, J. G.; VALLE DIAZ, J. D. ; HELGUERA FLEITAS, A.. **Jad at ehealth-kd challenge 2021: Simple neural network with bert for joint classification of key-phrases and relations**. In: PROCEEDINGS OF THE IBERIAN LANGUAGES EVALUATION FORUM (IBERLEF 2021), 2021.

[41] GUAN, Z.; LIU, R.. **Yunnan-deep at ehealth-kd challenge 2021: Deep learning model for entity recognition in spanish documents**. In: PROCEEDINGS OF THE IBERIAN LANGUAGES EVALUATION FORUM (IBERLEF 2021), 2021.

[42] YANG, M.. **Yunnan-1 at ehealth-kd challenge 2021: Deep-learning methods for entity recognition in medical text**. In: PROCEEDINGS OF THE IBERIAN LANGUAGES EVALUATION FORUM (IBERLEF 2021), 2021.

[43] HONNIBAL, M.; MONTANI, I.. **spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing**. To appear, 2017.

[44] NIVRE, J.; DE MARNEFFE, M.-C.; GINTER, F.; GOLDBERG, Y.; HAJIČ, J.; MANNING, C. D.; MCDONALD, R.; PETROV, S.; PYYSALO, S.; SILVEIRA, N.; TSARFATY, R. ; ZEMAN, D.. **Universal Dependencies v1: A multilingual treebank collection**. In: PROCEEDINGS OF THE TENTH INTERNATIONAL CONFERENCE ON LANGUAGE RESOURCES AND EVALUATION (LREC'16), p. 1659–1666, Portorož, Slovenia, May 2016. European Language Resources Association (ELRA).

[45] ARKHIPOV, M. V.; TROFIMOVA, M.; KURATOV, Y. ; SOROKIN, A.. **Tuning multilingual transformers for language-specific named entity recognition**. In: BSNLP@ACL, 2019.

[46] KENDALL, A.; GAL, Y. ; CIPOLLA, R.. **Multi-Task Learning Using Uncertainty to Weigh Losses for Scene Geometry and Semantics**. In: PROCEEDINGS OF THE IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION, p. 7482–7491, 2018.

[47] CAÑETE, J.; CHAPERON, G.; FUENTES, R.; HO, J.-H.; KANG, H. ; PÉREZ, J.. **Spanish pre-trained bert model and evaluation data**. In: PML4DC AT ICLR 2020, 2020.

[48] CAÑETE, J.. **Compilation of large spanish unannotated corpora**, May 2019.

[49] OTEGI, A.; AGIRRE, A.; CAMPOS, J. A.; SOROA, A. ; AGIRRE, E.. **Conversational question answering in low resource scenarios: A dataset and case study for basque**. In: PROCEEDINGS OF THE 12TH LANGUAGE RESOURCES AND EVALUATION CONFERENCE, p. 436–442, 2020.

[50] WU, Y.; SCHUSTER, M.; CHEN, Z.; LE, Q. V.; NOROUZI, M.; MACHEREY, W.; KRIKUN, M.; CAO, Y.; GAO, Q.; MACHEREY, K.; KLINGNER, J.; SHAH, A.; JOHNSON, M.; LIU, X.; KAISER, ; GOUWS, S.; KATO, Y.; KUDO, T.; KAZAWA, H.; STEVENS, K.; KURIAN, G.; PATIL, N.; WANG, W.; YOUNG, C.; SMITH, J.; RIESA, J.; RUDNICK, A.; VINYALS, O.; CORRADO, G.; HUGHES, M. ; DEAN, J.. **Google's neural machine translation system: Bridging the gap between human and machine translation**, 2016.

[51] CORTES, C.; VAPNIK, V.. **Support-vector networks**. Machine learning, 20(3):273–297, 1995.

[52] COLLOBERT, R.; WESTON, J.. **A unified architecture for natural language processing**. p. 160–167, 2008.

[53] COLLOBERT, R.; WESTON, J.; BOTTOU, L.; KARLEN, M.; KAVUKCUOGLU, K. ; KUKSA, P.. **Natural language processing (almost) from scratch**. Journal of machine learning research, 12(ARTICLE):2493–2537, 2011.

[54] YAO, L.; LIU, H.; LIU, Y.; LI, X. ; ANWAR, M. W.. **Biomedical Named Entity Recognition based on Deep Neutral Network**. International Journal of Hybrid Information Technology, 8(8):279–288, 2015.

[55] MIKOLOV, T.; CHEN, K.; CORRADO, G. ; DEAN, J.. **Efficient estimation of word representations in vector space**. 1st International Conference on Learning Representations, ICLR 2013 - Workshop Track Proceedings, p. 1–12, 2013.

[56] NGUYEN, T. H.; SIL, A.; DINU, G. ; FLORIAN, R.. **Toward Mention Detection Robustness with Recurrent Neural Networks**. 2016.

[57] MA, X.; HOVY, E.. **End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF**. 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016 - Long Papers, 2:1064–1074, 2016.

[58] KURU, O.; CAN, O. A. ; YURET, D.. **CharNER: Character-level named entity recognition**. COLING 2016 - 26th International Conference on Computational Linguistics, Proceedings of COLING 2016: Technical Papers, p. 911–921, 2016.

[59] DARWISH, K.. **Named entity recognition using cross-lingual resources: Arabic as an example**. ACL 2013 - 51st Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, 1:1558–1567, 2013.

[60] LIU, C.; SUN, W.; CHAO, W. ; CHE, W.. **Convolution neural network for relation extraction**. In: INTERNATIONAL CONFERENCE ON ADVANCED DATA MINING AND APPLICATIONS, p. 231–242. Springer, 2013.

[61] ZENG, D.; LIU, K.; LAI, S.; ZHOU, G. ; ZHAO, J.. **Relation classification via convolutional deep neural network**. In: PROCEEDINGS OF COLING 2014, THE 25TH INTERNATIONAL CONFERENCE ON COMPUTATIONAL LINGUISTICS: TECHNICAL PAPERS, p. 2335–2344, 2014.

[62] NGUYEN, T. H.; GRISHMAN, R.. **Relation extraction: Perspective from convolutional neural networks**. In: PROCEEDINGS OF THE 1ST WORKSHOP ON VECTOR SPACE MODELING FOR NATURAL LANGUAGE PROCESSING, p. 39–48, 2015.

[63] **The world's most spoken languages**. `https://www.aljazeera.com/news/2022/2/21/interactive-listen-to-the-best-sayings-in-25-different-languages`. Accessed: 2022-09-04.

[64] ZHANG, N.; JIA, Q.; DENG, S.; CHEN, X.; YE, H.; CHEN, H.; TOU, H.; HUANG, G.; WANG, Z.; HUA, N. ; CHEN, H.. **Alicg: Fine-grained and evolvable conceptual graph construction for semantic search at alibaba**. In: PROCEEDINGS OF THE 27TH ACM SIGKDD CONFERENCE ON KNOWLEDGE DISCOVERY AMP; DATA MINING, KDD '21, p. 3895–3905, New York, NY, USA, 2021. Association for Computing Machinery.

[65] LAHAV, D.; FALCON, J. S.; KUEHL, B.; JOHNSON, S.; PARASA, S.; SHOMRON, N.; CHAU, D. H.; YANG, D.; HORVITZ, E.; WELD, D. S. ; OTHERS. **A search engine for discovery of scientific challenges and directions**. AAAI, 2022.

[66] BRICKLEY, D.; BURGESS, M. ; NOY, N.. **Google dataset search: Building a search engine for datasets in an open web ecosystem**. In: THE WORLD WIDE WEB CONFERENCE, WWW '19, p. 1365–1375, New York, NY, USA, 2019. Association for Computing Machinery.

[67] WANG, H.; ZHAO, M.; XIE, X.; LI, W. ; GUO, M.. **Knowledge graph convolutional networks for recommender systems**. In: THE WORLD WIDE WEB CONFERENCE, p. 3307–3313, 2019.

[68] MAURO, N.; ARDISSONO, L.; COCOMAZZI, S. ; CENA, F.. **Information extraction for inclusive recommender systems**. In: 1ST WORKSHOP ON SOCIAL AND CULTURAL INTEGRATION WITH PERSONALIZED INTERFACES (SOCIALIZE), volumen 2903, p. 1–5. CEUR, 2021.

[69] BETANCOURT., Y.; ILARRI., S.. **Use of text mining techniques for recommender systems**. In: PROCEEDINGS OF THE 22ND INTERNATIONAL CONFERENCE ON ENTERPRISE INFORMATION SYSTEMS - VOLUME 1: ICEIS,, p. 780–787. INSTICC, SciTePress, 2020.

[70] WATANABE, K.. **Newsmap: A semi-supervised approach to geographical news classification**. Digital Journalism, 6(3):294–309, 2018.

[71] ZHANG, C.; GUPTA, A.; KAUTEN, C.; DEOKAR, A. V. ; QIN, X.. **Detecting fake news for reducing misinformation risks using analytics approaches**. European Journal of Operational Research, 279(3):1036–1052, 2019.

[72] CHOUDHARY, A.; ARORA, A.. **Linguistic feature based learning model for fake news detection and classification**. Expert Systems with Applications, 169:114171, 2021.

[73] LI, J.; SUN, A.; HAN, J. ; LI, C.. **A survey on deep learning for named entity recognition**. IEEE Transactions on Knowledge and Data Engineering, 34(1):50–70, 2020.

[74] LIU, K.. **A survey on neural relation extraction**. Science China Technological Sciences, 63(10):1971–1989, 2020.