UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE ENGENHARIA DE COMPUTAÇÃO

NICOLAS EYMAEL DA SILVA

Extraction of entities and relations in Portuguese from the Second HAREM Golden Collection

Work presented in partial fulfillment
of the requirements for the degree of
Bachelor in Computer Engineering

Advisor: Prof. Dr. Dante Augusto Couto Barone
Coadvisor: MSc. Eduardo Gabriel Cortes

Porto Alegre
May 2021

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
Reitor: Prof. Carlos André Bulhões Mendes
Vice-Reitora: Prof^{a}. Patricia Helena Lucas Pranke
Pró-Reitoria de Ensino (Graduação e Pós-Graduação): Prof^{a}. Cíntia Inês Boll
Diretora do Instituto de Informática: Prof^{a}. Carla Maria Dal Sasso Freitas
Diretora da Escola de Engenharia: Prof^{a}. Carla Schwengber Ten Caten
Coordenador do Curso de Engenharia de Computação: Prof. Walter Fetter Lages
Bibliotecária-chefe do Instituto de Informática: Beatriz Regina Bastos Haro
Bibliotecária-chefe da Escola de Engenharia: Rosane Beatriz Allegretti Borges

”Little by little, what you’ve begun will naturally become important to you.
What you need at the start is a little bit of curiosity.”

— HARUICHI FURUDATE

ACKNOWLEDGMENTS

First of all, I want to thank the colleagues who accompanied me during this long academic journey. I am sure that many of them will continue to be part of my life.

I thank my advisor Dante and coadvisor Eduardo for their assistance in developing this work and for their willingness to share knowledge.

I also want to thank my friends who have been part of my daily life since school. I look forward to the end of the quarantine so that we can meet again.

Finally, I thank my family, especially my father and my mother, who always encouraged me to study and supported me unconditionally in every moment of my life. If I made it this far, I owe it to you.

ABSTRACT

Information Extraction is an essential process for automatically building a Knowledge Graph, a type of knowledge base that represents knowledge through semantic connections and has been gaining focus in recent years. Two tasks required during this construction are Named Entity Recognition (NER), responsible for identifying and classifying the entities in the text, and Relation Extraction (RE), responsible for identifying and classifying the relations between these entities. These two tasks combined will generate the tuples that form the Knowledge Graph. Although there are already works that deal with these two tasks, many of them are focused on the English language and few on Portuguese. The goal of this work was the development of machine learning models capable of extracting entities and relations from texts in Portuguese. The first model was used to extract entities through the Simple Transformers library, while the second model was used to determine the relations between entities through the Kindred library. Both models were trained and evaluated using a simplified version of the Second HAREM Golden Collection dataset, a golden standard for NLP in Portuguese. After evaluating the models, it was observed that the results obtained in the NER task were good for the main classes present in the dataset, however, the results of the RE task did not meet expectations and the metrics were lower compared to the related works. Finally, it would be interesting to develop new models for the RE task using the spaCy or Transformers libraries, alternatives that are more complex than Kindred, but more effective.

Keywords: Named Entity Recognition. Relation Extraction. HAREM. Knowledge Graph.

Extração de entidades e relações em português a partir da Coleção Dourada do Segundo HAREM

RESUMO

A Extração de Informações é um processo essencial para construir um Grafo de Conhecimento de forma automatizada, um tipo de base de conhecimento que representa o conhecimento através de conexões semânticas e que vem ganhando foco nos últimos anos. Duas tarefas necessárias durante essa construção são o Reconhecimento de Entidades Nomeadas (REN), responsável por identificar e classificar as entidades do texto, e a Extração de Relações (ER), responsável por identificar e classificar as relações entre essas entidades. Essas duas tarefas combinadas irão gerar as tuplas que formam o Grafo de Conhecimento. Apesar de já existirem trabalhos que tratam dessas duas tarefas, muitos deles são voltados para a língua inglesa e poucos para o português. O objetivo deste trabalho foi o desenvolvimento de modelos de aprendizado de máquina capazes de extrair entidades e relações de textos em português. O primeiro modelo foi utilizado para a extração das entidades por meio da biblioteca Simple Transformers, enquanto que o segundo modelo foi utilizado para determinar as relações entre as entidades através da biblioteca Kindred. Ambos os modelos foram treinados e avaliados utilizando uma versão simplificada do conjunto de dados do Segundo HAREM, um padrão de ouro para o Processamento de Linguagem Natural em português. Após a avaliação dos modelos, observou-se que os resultados obtidos na tarefa de REN foram bons para as principais classes presentes no conjunto de dados, no entanto os resultados da tarefa de ER não atenderam às expectativas e as métricas foram inferiores se comparadas aos trabalhos relacionados. Por fim, seria interessante desenvolver novos modelos para a tarefa de ER utilizando as bibliotecas spaCy ou Transformers, alternativas que são mais complexas do que o Kindred, porém mais eficazes.

Palavras-chave: Reconhecimento de Entidades Nomeadas, Extração de Relações, HAREM, Grafo de Conhecimento.

LIST OF ABBREVIATIONS AND ACRONYMS

AI	Artificial Intelligence
BERT	Bidirectional Encoder Representations from Transformers
BIO	Beginning-Inside–Outside
CRF	Conditional Random Fields
EL	Entity Linking
GC	Golden Collection
IE	Information Extraction
KG	Knowledge Graph
LG	Local Grammars
ML	Machine Learning
NED	Named Entity Disambiguation
NER	Named Entity Recognition
NLP	Natural Language Processing
OIE	Open Information Extraction
QA	Question Answering
RE	Relation Extraction
ST	Simple Transformers

LIST OF FIGURES

LIST OF TABLES

CONTENTS

1 INTRODUCTION

Information Extraction (IE) is an important task in natural language processing and text mining, which consists of extracting structured information from unstructured or semi-structured texts (JIANG, 2012). This information can be presented to the user or even used to improve other systems, such as search engines.

The first IE systems worked using rule-based models, that is, the information was identified using linguistic patterns developed by humans. These systems can achieve good performance, but the process of creating rules manually is laborious and the rules are highly domain-dependent. Because of these limitations, the researchers decided to approach this task through statistical machine learning models.

A task derived from IE is Open Information Extraction (OIE). OIE systems seek to extract all important information, regardless of the domain, from a large and diverse corpus. This information can be useful entities and relations, which are usually represented by tuples. Finally, these tuples can be used in the construction of databases, such as a Knowledge Graph (KG) (MUHAMMAD et al., 2020). This KG can then be used in Financial Analytics, Question Answering, and other applications.

To generate these tuples, two IE subtasks are required: Named Entity Recognition (NER) and Relation Extraction (RE). The NER is responsible for identifying and classifying the entities present in the texts, while the RE identifies the existing relations between these entities. Although many works deal with these tasks, they are usually focused on the English language and not on Portuguese, both for the NER (CASTRO; SILVA; SOARES, 2018) and the RE (ABREU; VIEIRA, 2017).

Among the works focused on Portuguese, the one that stands out the most is HAREM. HAREM is an evaluation contest organized by Linguateca, which aims to carry out the evaluation of NER and RE systems for the Portuguese language (MOTA; SANTOS, 2008). The HAREM corpus is a reference in the NLP area of the Portuguese community and is characterized by having a large set of texts annotated and validated by humans.

Therefore, the goal of this work is the development of computational models capable of extracting entities and relations in the Portuguese language. These models will be trained and evaluated using a simplified version of the Second HAREM corpus. After the executions, the results of each task will be analyzed and compared with other related systems.

The rest of this work is organized as follows. Chapter 2 describes the concepts used to carry out and understand this research. Chapter 3 provides an overview of related work. Chapter 4 presents the procedures and tools adopted at each stage of this work. Chapter 5 presents an analysis of the results obtained in the experiments. Finally, in chapter 6 the conclusions are presented.

2 THEORETICAL BACKGROUND

In order to provide the necessary theoretical background, this chapter describes the main concepts covered in this work. In total, this chapter contains 7 sections. Section 2.1 briefly introduces the NLP area. Section 2.2 presents the IE task. Sections 2.3 and 2.4 describe two IE subtasks: EL and RE, respectively. Section 2.5 explains what a KG is. Section 2.6 presents the metrics used in the evaluation of the models. Finally, section 2.7 explains the cross-validation method.

2.1 Natural Language Processing

Natural Language Processing (NLP) is an Artificial Intelligence (AI) research area that explores how computers can be used to understand and manipulate texts in natural language (CHOWDHURY, 2003).

The goal of NLP researchers is to gather knowledge of how humans use language so that it is possible to develop appropriate techniques and tools so that computers can understand and perform tasks related to human language.

The field of NLP began in the 1940s. The first works in the area approached the problem in a very simplistic way, taking into account only the ordering of a dictionary of words allowed by language (LIDDY, 2001). This approach produced poor results and the researchers realized that this task is much more difficult than they imagined.

In recent years, the field was growing rapidly. This is mainly due to three factors: the increasing volume of texts available digitally, the development of computers with increasing speed and memory, and the advent of the Internet. The focus of researchers today is to develop systems that achieve good results with general text, taking into account the variability and ambiguity of language.

2.2 Information Extraction

Information Extraction (IE) refers to the automatic extraction of structured information, such as entities and relations between these entities, from unstructured or semi-structured documents. With roots in the NLP community, the topic of IE now engages many different communities spanning machine learning (ML), information retrieval, database, web, and document analysis (SARAWAGI, 2008).

Traditional approaches to IE focus on answering well-defined requests over a pre-defined set of relations on small and homogeneous corpora (NIKLAUS et al., 2018). These systems worked through rule-based models that used linguistic patterns annotated manually. Despite achieving good performance, the process of creating the rules was laborious and highly domain-dependent.

To reduce the manual effort required by IE approaches, a new extraction paradigm was introduced: Open Information Extraction (OIE) (ETZIONI et al., 2008). Unlike traditional IE methods, OIE systems seek to extract all types of relations, regardless of the domain. In that way, it assists the domain-independent discovery of relations from large and heterogeneous corpora such as the Web. They have been used for a wide variety of applications, such as textual entailment, question answering, and knowledge base population (STANOVSKY et al., 2018).

2.3 Entity Linking

Entity Linking (EL) refers to the task of uniquely identifying each entity mentioned in an unstructured text and linking the mentions with the corresponding entities in a knowledge base (SHEN; WANG; HAN, 2014).

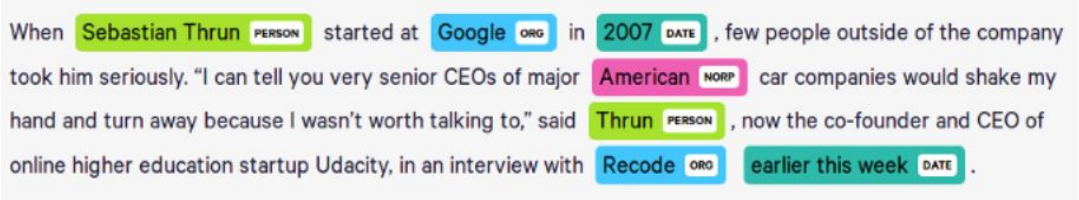
A named entity is a physical or abstract object that can be identified by a proper name. These objects are classified as people, places, organizations, works, etc. In addition, numeric and temporal expressions can also be considered named entities.

The first stage of EL is the Named Entity Recognition (NER). This step is responsible for identifying the occurrences of named entities in a text and classifying them according to pre-defined categories (NADEAU; SEKINE, 2007).

Figure 2.1 shows the result of the NER task applied to a paragraph of a text. The NER can identify, for example, that “Sebastian Thrun” is an entity of type Person and that “Google” is an entity of type Organization. However, the NER is unable to conclude that the entities “Sebastian Thrun” and “Thrun” actually refer to the same person.

In order to make this connection between the entity identified in the text and the entity present in the knowledge base, it is necessary to perform the second stage of the EL, called the Named Entity Disambiguation (NED) (HOFFART et al., 2011). The NED task is responsible for linking the entity correctly to the knowledge base, both in cases where a word has several different meanings and in cases where different words have the

Figure 2.1: NER applied to a paragraph of a text.



When Sebastian Thrun PERSON started at Google ORG in 2007 DATE , few people outside of the company took him seriously. “I can tell you very senior CEOs of major American NORP car companies would shake my hand and turn away because I wasn’t worth talking to,” said Thrun PERSON , now the co-founder and CEO of online higher education startup Udacity, in an interview with Recode ORG earlier this week DATE .

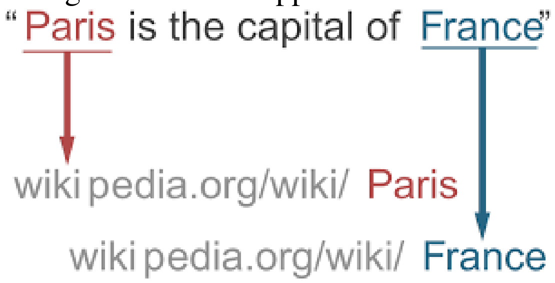
Source: AI Time Journal

same meaning (which is the case with “Thrun” in the example).

Figure 2.2 shows the result of the NED task applied to a sentence. The entity “Paris” has more than one correspondence in the knowledge base, as it can be related both to the capital of France and to a small city in Arkansas, USA. Likewise, the entity “France” can also refer to the French football team, for example.

It is the responsibility of the NED to make this differentiation of meanings and correctly relate the entities to the corresponding entities in the knowledge base. In this case, “Paris (the city in France) is the capital of France (the country in Europe)”.

Figure 2.2: NED applied to a sentence.


“Paris is the capital of France”
wikipedia.org/wiki/ Paris
wikipedia.org/wiki/ France

Source: Wikipedia

Most EL works in the literature address NER and NED independently. Thus, it is possible to obtain great results with an accuracy of up to 99% (RAIMAN; RAIMAN, 2018). The works that modeled an end-to-end system with both stages managed to obtain good results, but still do not compare to approaches with independent methods (KOLITSAS; GANEA; HOFMANN, 2018; HULST et al., 2020; PICCINNO; FERRAGINA, 2014).

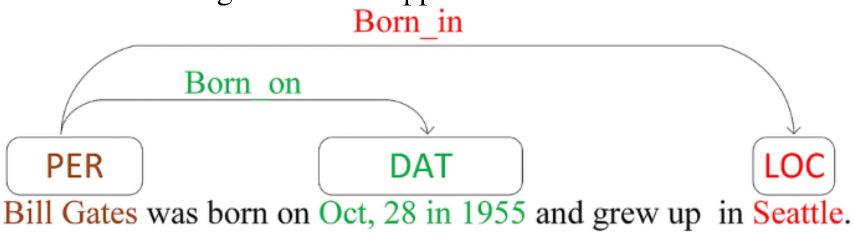
2.4 Relation Extraction

Relation Extraction (RE) is the task of predicting attributes and relations between entities in a sentence. This task is crucial for many NLP applications, especially for building knowledge graphs (HUANG; WANG, 2017).

Relations occur between two or more entities, but not necessarily of the same type. In addition to detecting whether or not there is any type of relation between entities, it is still necessary to determine in which class that relation fits. An entity of type Person, for example, can have a semantic relation with a Local (like the relation “was born in”) or even with another Person (like the relation “is the daughter of”).

Figure 2.3 shows the result of the RE task applied to a sentence. The entity “Bill Gates” has a “born in (time)” relation with the entity “Oct, 28 in 1955”. In addition, the same entity “Bill Gates” also has another relation, this time “born in (local)”, with the entity “Seattle”. Although “Oct, 28 in 1955” and “Seattle” are indirectly linked through “Bill Gates”, the two entities have no relation to each other.

Figure 2.3: ER applied to a sentence.


Born_in
Born_on
PER
DAT
LOC
Bill Gates was born on Oct, 28 in 1955 and grew up in Seattle.

Source: (ZHANG et al., 2020)

The RE task can be approached in different ways. There are supervised methods that use previous annotations, unsupervised methods that are based on generic extraction patterns, or even semi-supervised methods that apply concepts from the other two methods. Unfortunately, a problem common to all methods is the shortage of models aimed at the Portuguese language (ABREU; BONAMIGO; VIEIRA, 2013), mainly due to the lack of annotated data.

On the other hand, there is already a variety of work in English using different methods and datasets. Models using distantly supervised extraction applied to New York Times texts can achieve an accuracy of around 80 to 85% (XU; BARBOSA, 2019; WU; FAN; ZHANG, 2019; YE; LING, 2019).

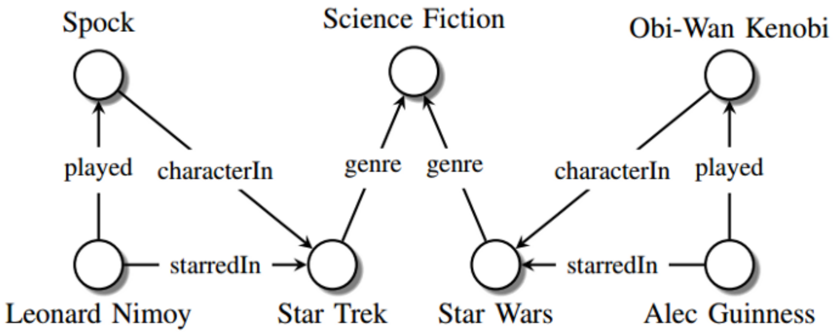
2.5 Knowledge Graph

A Knowledge Graph (KG) is a type of knowledge base that integrates data using a graph structure and has been gaining focus on AI and NLP applications, such as Question Answering systems, for example. A KG is a collection of relational facts that are often represented by tuples (WANG et al., 2014).

A key feature of a KG is that each entity needs to be connected to another entity, that is, the definition of an entity always includes another entity. These connections are what make up the graph.

Figure 2.4 shows a KG with entities referring to science fiction films. Another feature of KG is that it is expandable, that is, it is possible to add a new entity (the actor "Ewan McGregor", for example) and relate it to the other entities that already exist in the graph (such as a "played" relation with the entity "Obi-Wan Kenobi"). In addition, this expansion process can even be automated (YOO; JEONG, 2020).

Figure 2.4: KG with entities related to films.



Source: (NICKEL et al., 2015)

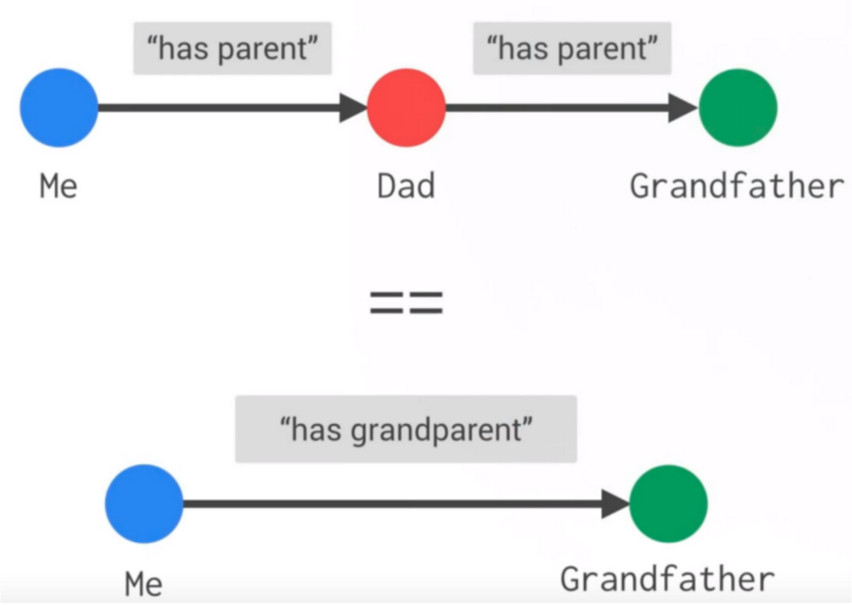
The elementary unit of a KG is the subject-predicate-object tuple (NICKEL et al., 2015). In any graph, each tuple defines a connection between two nodes. In the case of a KG, connections are about relations and nodes are entities. After processing the EL and RE techniques together on texts, it is possible to obtain the tuple collection necessary to build a KG.

In addition, the KG is also capable of inferring relations between entities (LIU et al., 2016). In Figure 2.5, an example of the inference of the relation between the entities "Me" and "Grandfather" is presented. Even if in the original graph the two entities are not directly connected, it is possible to make this connection through the "Dad" entity.

2.6 Evaluation metrics

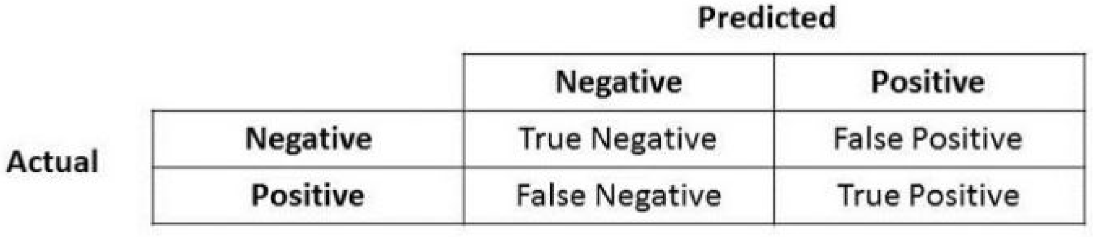
The performance of a classification model can be evaluated based on a confusion matrix (see Figure 2.6). In this matrix, the row represents the current class, while the column represents the predicted class.

Figure 2.5: Inference of relations between entities.



Source: The Author

Figure 2.6: Confusion Matrix for Binary Classification.



		Predicted	
		Negative	Positive
Actual	Negative	True Negative	False Positive
	Positive	False Negative	True Positive

Source: Towards Data Science

From this confusion matrix, TP and TN denote the number of positive and negative instances that are correctly classified. Meanwhile, FP and FN denote the number of misclassified negative and positive instances, respectively. Instances can be entities or relations that were extracted from a text, for example (DALIANIS, 2018).

Among the metrics that can be calculated from the matrix, those used in this work are precision, recall, and F-score. The formulas for each of these metrics can be seen in Figure 2.7.

Precision is used to measure the positive patterns that are correctly predicted from the total predicted patterns in a positive class. Recall is used to measure the fraction of positive patterns that are correctly classified. Finally, F-score represents the harmonic mean between recall and precision values (HOSSIN; SULAIMAN, 2015).

When it comes to multi-class cases, metrics may have two different strategies: MICRO-level approach and MACRO-level approach. Figure 2.8 shows the formulas of

Figure 2.7: Formulas to calculate the metrics used in this work.

$$\text{Precision} = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

$$\text{Recall} = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

$$\text{F1} = 2 \text{ x } \frac{Precision * Recall}{Precision + Recall}$$

Source: Towards Data Science

the two strategies using the precision metric as an example. In both formulas, K is the number of classes present in the dataset.

Figure 2.8: Differences in calculating precision between a MACRO-level and MICRO-level approach.

$$\begin{matrix}\text{MACRO}\\ \text{PRECISION}\end{matrix} = \frac{\sum_{k=1}^{K} \text{PRECISION}_k}{K}$$

$$\begin{matrix}\text{MICRO}\\ \text{PRECISION}\end{matrix} = \frac{\sum_{k=1}^{K} \text{TP}_k}{\sum_{k=1}^{K} \text{TP}_k + \sum_{k=1}^{K} \text{FP}_k}$$

Source: The Author

The MACRO approach considers all the classes as basic elements of the calculation: each class has the same weight in the average so that there is no distinction between highly and poorly populated classes. This implies that the effect of the biggest classes has the same importance as small ones have. On the other hand, the idea of the MICRO approach is to consider all the units together, without taking into consideration possible differences between classes. It means that it gives more importance to big classes because it just considers all the units together. Poor performance on small classes is not so important, since the number of units belonging to those classes is small compared to the dataset size (GRANDINI; BAGLI; VISANI, 2020).

2.7 Cross-validation

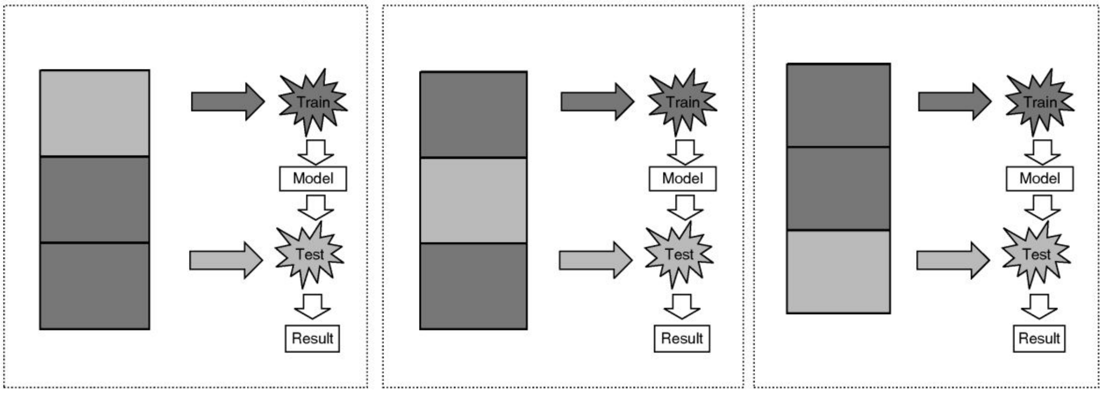
Cross-validation is a data resampling method to evaluate the generalization ability of predictive models and to prevent overfitting. This method is widely used to estimate the true prediction error of models and to tune model parameters (BERRAR, 2019). The purpose of cross-validation is to provide an estimate for the performance of the model on new data.

The method consists of evaluating and comparing learning algorithms by dividing data into two segments: one used to learn or train a model and the other used to validate the model. The basic form of cross-validation is k-fold cross-validation.

In k-fold cross-validation, the data is first partitioned into k equally (or nearly equally) sized segments or folds. Subsequently, k iterations of training and validation are performed such that within each iteration a different fold of the data is held out for validation while the remaining k-1 folds are used for learning. In data mining and machine learning 10-fold cross-validation (k = 10) is the most common (REFAEILZADEH; TANG; LIU, 2009). This value for k has been found through experimentation to generally result in a model with low bias and a modest variance (JAMES et al., 2013).

Figure 2.9 illustrates how the method works with k = 3. In each iteration, the darker folds are used to train the model, while the lighter fold is used to evaluate its performance. The metrics obtained in the results of the three executions can be aggregated by calculating the average.

Figure 2.9: Procedure of three-fold cross-validation.



Source: (REFAEILZADEH; TANG; LIU, 2009)

3 RELATED WORK

In this chapter, the works related to the tasks of NER and RE using the Second HAREM are presented. The research of the works was carried out through Google Scholar. The methods and results of each work are described below.

At the Second HAREM Workshop, organized by Linguateca in 2008, 10 systems were participating in the identification and classification of named entities (MOTA; SANTOS, 2008). Each participant performed 1 to 4 runs with different scenarios. The systems that obtained the best metrics in the identification of entities in the total scenario, that is, encompassing all classes of the dataset, are shown in Table 3.1.

Table 3.1: The best systems participating in the task of identifying entities.

Metric	System	Value
F-score	Priberam	71.0976%
Precision	SEIGeo	90.5599%
Recall	Priberam	72.2906%

Source: Adapted from Mota and Santos (2008)

Table 3.2 presents the best systems for classifying entities in the total scenario. The purpose of identifying entities is only to find the entities present in the text, while the classification of entities also involves finding out in which category the entity fits.

Table 3.2: The best systems participating in the task of classifying entities.

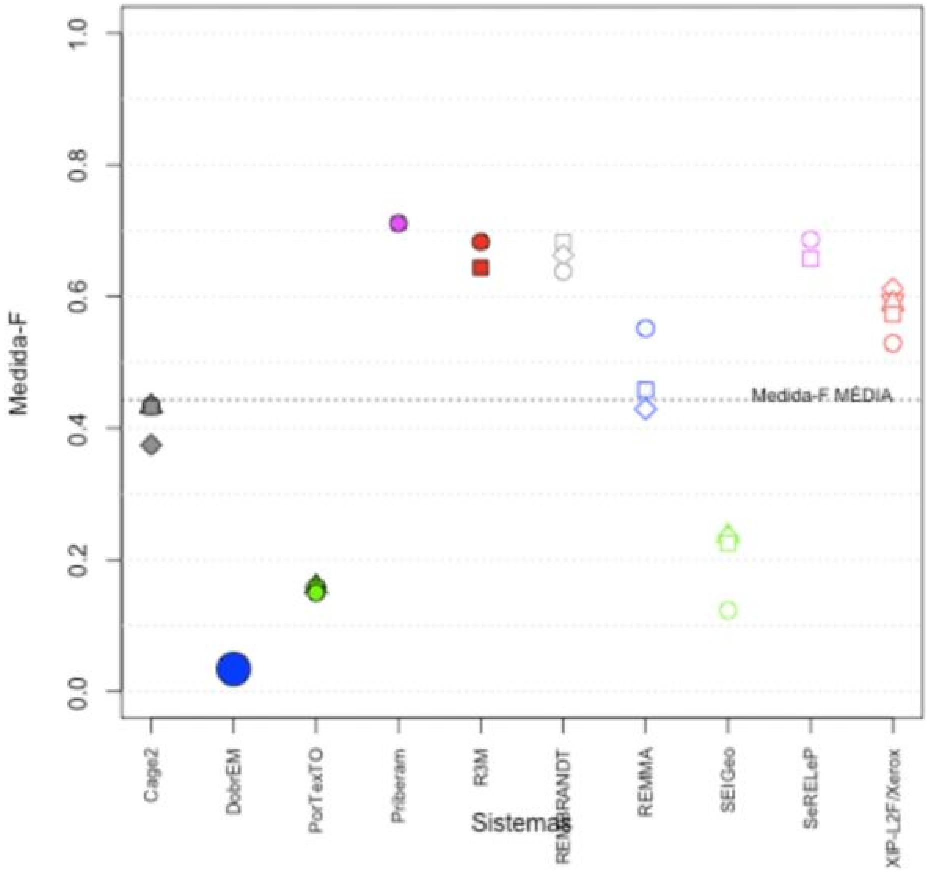
Metric	System	Value
F-score	Priberam	57.11386%
Precision	SeRELeP	81.79828%
Recall	Priberam	51.45506%

Source: Adapted from Mota and Santos (2008)

Figures 3.1 and 3.2 show the F-score of each of the systems participating in the task of identifying and classifying entities, respectively. Among all the participants, it is noted that the Priberam (AMARAL et al., 2008) and REMBRANDT (CARDOSO, 2008) systems are the ones that obtained the best performances, taking into account both tasks.

Both the Priberam system and the REMBRANDT (as well as most of the participating systems) use a set of rules and clauses generated manually in combination with dictionaries and ontologies. This shows that the community dedicated to NER in Portuguese at the time preferred language approaches and had not embraced machine learning techniques, contrary to the situation for English (FREITAS et al., 2010).

Figure 3.1: F-score of the systems participating in the entity identification task.



Source: (MOTA; SANTOS, 2008)

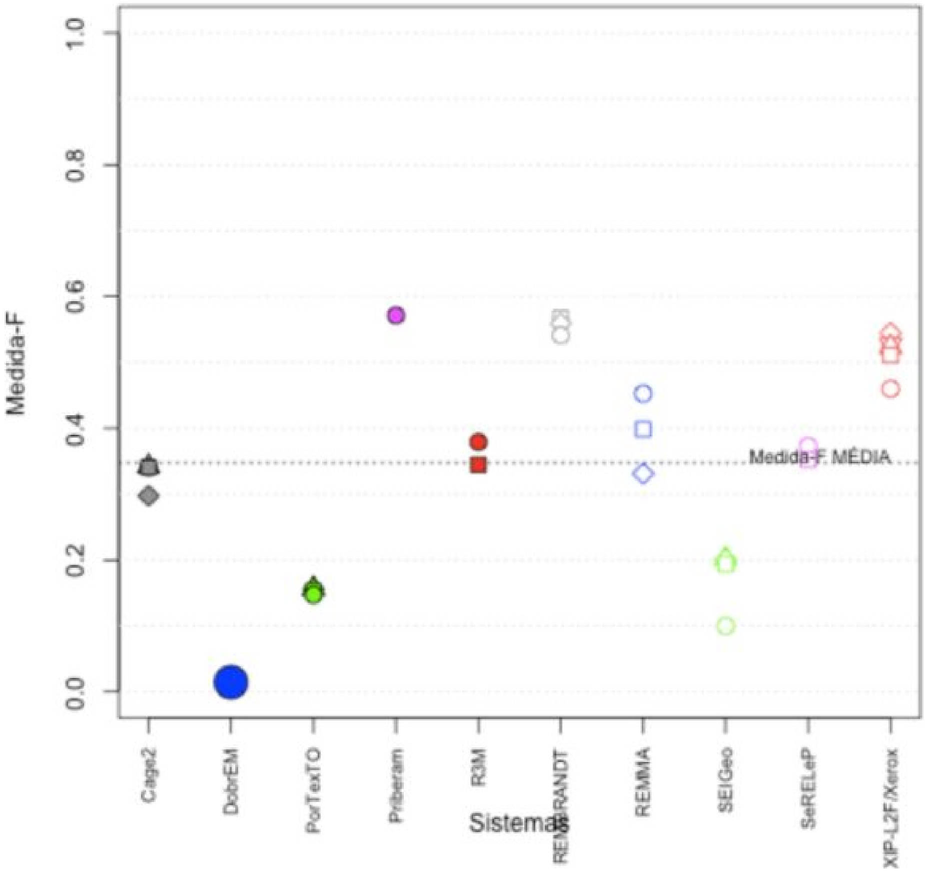
In addition to the systems participating in the Workshop, other NER models also used the Second HAREM dataset in the following years.

The NERP-CRF system (AMARAL; VIEIRA, 2014), unlike the systems mentioned above, uses a machine learning model based on Conditional Random Fields (CRF) for the task of entity classification. The NERP-CRF showed better results than the participating systems for the metrics of precision (83.48%) and F-score (57.92%).

However, this result is biased because a single corpus, the golden collection (GC) of the Second HAREM, was used for training and test, through cross-validation. To make the comparison fairer with the participating systems, a second run was performed using the First HAREM GC as training and the Second HAREM GC as a test. This second run showed slightly worse results than the first, with an accuracy of 80.77% and an F-score of 48.43%.

The CRF+LG system (PIROVANI; OLIVEIRA, 2018) is a hybrid system that uses linguistic methods and machine learning approaches in the NER task. The CRF+LG combines labeling obtained by Conditional Random Fields (CRF) with a term classification

Figure 3.2: F-score of the systems participating in the entity classification task.



Source: (MOTA; SANTOS, 2008)

obtained from Local Grammars (LGs). The experiments were performed using the First HAREM GC for training and the Second HAREM GC for testing. The results obtained from the experiments indicate an F-score of 70.62% and 57.8% in the identification and classification of entities, respectively.

Another system worth mentioning is the BERT-CRF, proposed by Souza, Nogueira and Lotufo (2019). This NER system combines the transfer capabilities of BERT with the structured predictions of CRF. However, the experiments were carried out using only the First HAREM, which makes it difficult to compare the results with the other works already mentioned.

Table 3.3 summarizes the relevant information on the systems related to the entity classification task, while Table 3.4 presents the metrics obtained for the entity identification task. The NERP-CRF system did not provide metrics for the task of identifying entities.

Among the participants of the Workshop, only the SeRELeP (BRUCKSCHEN et

Table 3.3: NER systems and results of entity classification.

System	*Year*	*Approach*	*Precision*	*Recall*	*F-score*
Priberam	2008	Linguistic method with manually generated rules	64.17%	51.46%	57.11%
REMBRANDT	2008	Linguistic method with manually generated rules in combination with Wikipedia/DBpedia pages	64.97%	50.36%	56.74%
NERP-CRF (v1)	2014	ML with CRF (just Second HAREM)	**83.48%**	44.35%	**57.92%**
NERP-CRF (v2)	2014	ML with CRF (First HAREM and Second HAREM)	80.77%	34.59%	48.43%
CRF+LG	2018	Hybrid system with CRF and LG	65.46%	**51.75%**	57.8%

Source: The Author

Table 3.4: Results of entity identification.

System	*Precision*	*Recall*	*F-score*
Priberam	69.94%	**72.29%**	**71.10%**
REMBRANDT	75.77%	62.14%	68.28%
CRF+LG	**78.58%**	64.12%	70.62%

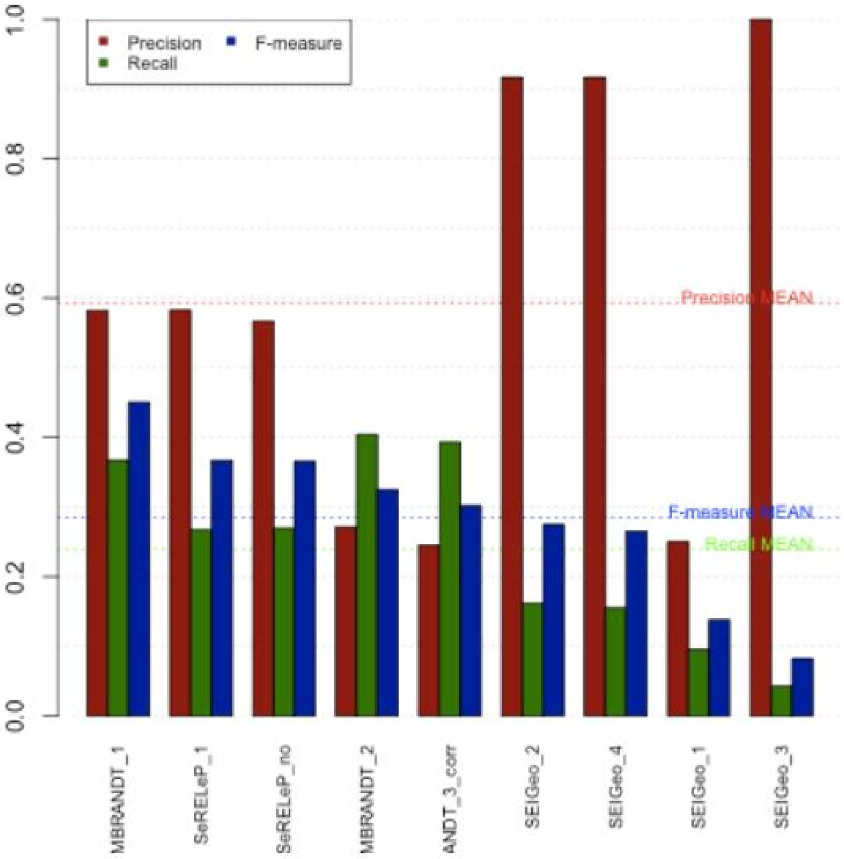
Source: The Author

al., 2008), SEIGeo (CHAVES, 2008) and REMBRANDT systems performed the task of extracting relations. This step used the ReRelEM dataset, a subset of the Second HAREM dataset that has annotated relations. The classes of relations are “identity”, “inclusion”, “locality” and “other”.

In Figure 3.3, it is possible to view the metrics obtained by these 3 systems. However, it is not possible to directly compare their performances. This is because each system approached the task of extracting relations in a different way. The SeRELeP system did not attempt to classify relations with the “other” class, while the SEIGeo system focused only on relations with the “inclusion” class. The only system that tried to classify all possible relations was REMBRANDT.

Another work related to extracting relations that also used the HAREM dataset is the RelP system (ABREU; VIEIRA, 2017). RelP extracts any descriptor that describes a relation between named entities in the organization domain by applying the CRF method. The metrics obtained by RelP were an accuracy of 51% and an F-score of 43% for the

Figure 3.3: Metrics of the systems participating in the RE task.



Source: (FREITAS et al., 2009)

case of exact matching, and an accuracy of 67% and an F-score of 58% for the case of partial matching. However, it is difficult to make a comparison with the other HAREM systems because the dataset used is not the same. The dataset used by RelP is a subset of HAREM that has only relations related to entities of type "organization" (manually annotated).

4 METHODOLOGY

This chapter presents the procedures and tools adopted in the elaboration of this work. The work was developed using the Python programming language since it has several libraries for data manipulation and ML, and the code is available on GitHub[1].

Section 4.1 presents the dataset used in this work and the preprocessing steps performed on it. Sections 4.2 and 4.3 describe the experiments related to the tasks of NER and RE, respectively.

4.1 Dataset and Preprocessing

The dataset used for both training and testing the system was the Second HAREM GC with manually annotated relations. It is important to note that this dataset was made available by Linguateca in April 2010, so it is not the same as the one used at the Second HAREM Workshop in September 2008.

At the Workshop, the tasks of NER and RE were carried out separately with different datasets. The ReRelEM GC, used in the RE task, was a subset of the Second HAREM GC, used in the NER task. This subset contained manually annotated entities and relations, whereas the complete set contained only entity annotations.

Two years after the Workshop, Linguateca made available a new version of the Second HAREM GC that includes the annotation of all existing relations between entities throughout the dataset. This new version has 7846 registered entities and 4847 relations between the entities. The distribution of entities and relations can be seen in Tables 4.1 and 4.2, respectively. It is important to note that some entities are classified in more than one category.

Figure 4.1 presents an excerpt from the dataset containing some entities and relations. The file made available by Linguateca is in XML format in which each element has a tag. The EM tag indicates that the word (or words) is an entity named with a category (CATEG), type (TYPE), and subtype (SUBTYPE), all of which are optional.

The other two attributes, also optional, are COREL and TIPOREL. COREL informs which entities have a relation with the entity in question, while TIPOREL informs the types of each of these relations. If the entity has more than one relation, they are separated by a space.

[1] <www.github.com/NicolasEymael/NER-RE-SecondHAREM>

Table 4.1: Distribution of entities in the dataset. Some entities have more than one category.

Entity category	#
ABSTRACCAO	439
ACONTECIMENTO	368
COISA	388
LOCAL	1608
OBRA	552
ORGANIZACAO	1260
OUTRO	112
PESSOA	2240
TEMPO	1206
VALOR	356

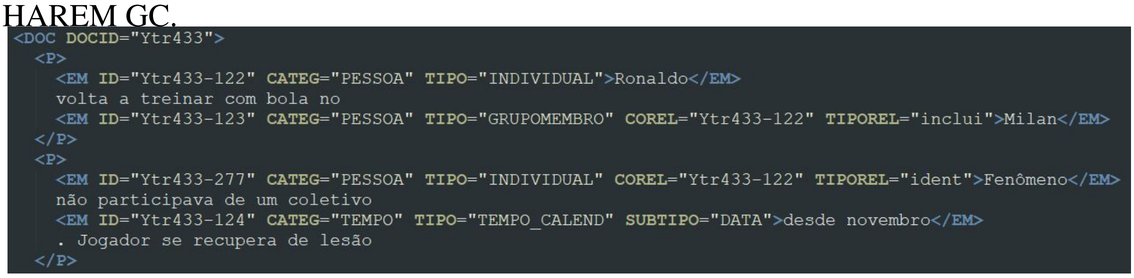
Source: The Author

The P tag indicates that the excerpt is a sentence that may or may not contain entities and the DOC tag informs which document these sentences are in. It is important to note that entities can only have relations with other entities contained in the same document.

In the example, two sentences in a document were highlighted. The first is the sentence “Ronaldo volta a treinar com bola no Milan”, which has two entities: “Ronaldo” and “Milan”. The entity “Ronaldo” has the category PESSOA and the type INDIVIDUAL, while the entity “Milan” shares the same category but has the type GRUPOMEMBRO. In addition, the entity “Milan” has a relation “inclui” with the entity “Ronaldo”.

The second sentence is “Fenômeno não participava de um coletivo desde novembro. Jogador se recupera de lesão”, which also has two entities. The first is the entity “Fenômeno”, which has the category PERSON, the type INDIVIDUAL, and the relation “ident” with the entity “Ronaldo”. The other entity is “desde novembro”, which has the category TEMPO, the type TEMPO_CALEND, and the subtype DATA.

Figure 4.1: Excerpt from the XML file showing the annotations present in the Second HAREM GC.



```
<DOC DOCID="Ytr433">
  <P>
    <EM ID="Ytr433-122" CATEG="PESSOA" TIPO="INDIVIDUAL">Ronaldo</EM>
    volta a treinar com bola no
    <EM ID="Ytr433-123" CATEG="PESSOA" TIPO="GRUPOMEMBRO" COREL="Ytr433-122" TIPOREL="inclui">Milan</EM>
  </P>
  <P>
    <EM ID="Ytr433-277" CATEG="PESSOA" TIPO="INDIVIDUAL" COREL="Ytr433-122" TIPOREL="ident">Fenômeno</EM>
    não participava de um coletivo
    <EM ID="Ytr433-124" CATEG="TEMPO" TIPO="TEMPO_CALEND" SUBTIPO="DATA">desde novembro</EM>
    . Jogador se recupera de lesão
  </P>
```

Source: The Author

Due to the complexity of classifying each entity by category, type, and subtype

Table 4.2: Distribution of relations in the dataset.

Relation type	#
autor_de	55
causador_de	22
consequencia_de	1
data_de	97
data_morte	10
data_nascimento	6
datado_de	7
ident	2265
inclui	357
incluido	508
local_morte	4
local_nascimento_de	46
localizacao_de	4
localizado_em	27
natural_de	57
nome_de	5
nome_de_data_de	2
nome_de_ident	28
nome_de_inclui	3
nome_de_incluido	2
nome_de_obra_de	6
nome_de_outrarel	3
nome_de_pratica_se	1
nome_de_praticado_por	1
nome_de_vinculo_inst	1
nomeado_por	6
obra_de	87
ocorre_em	103
outra_edicao	3
outrarel	101
participante_em	89
periodo_vida	5
personagem_de	14
pratica_se	16
praticado_em	67
praticado_por	16
praticante_de	27
produtor_de	32
produzido_por	22
propriedade_de	17
proprietario_de	21
relacao_familiar	88
relacao_profissional	17
residencia_de	16
residente_de	3
sede_de	250
ter_participacao_de	64
vinculo_inst	265

Source: The Author

(see Table 4.3), a new classification system was created with a reduced set of classes. The Beautiful Soup and pandas libraries were used to convert the XML file to a Dataframe, to facilitate data manipulation.

Table 4.4 shows the distribution of the new inferred classes, as well as the rules that were used to map the old classes to the new ones. Most of the rules just directly mapped the old category to the new class, except for the classes INDIVIDUO, ORGANIZACAO, and OUTRO. The INDIVIDUO class was assigned to entities that represent a single person, such as “Barack Obama” or “the pope”, for example. The class ORGANIZACAO was assigned to entities that represent a group of people, such as companies, institutions, and teams. Among the entities classified as ORGANIZACAO, some examples are “Google”, “FC Porto”, “European Union”, and “the Beatles”.

Finally, the class OUTRO includes all entities that could not be classified in the other classes. In addition, the entities that had the category COISA were also classified as

OUTRO. This is due to the similarity observed between these entities, such as the entity "Internet", which had the category OUTRO, and the entity "PowerPoint", which had the category COISA. The total of 7817 entities instead of 7846 is because the entities that did not have the CATEG attribute were discarded.

In addition to changes in entity types, some relation types have also been changed. The first change was to remove the additional information that was in some relations. For example, the entity "Aragão" had a relation of type "LOCAL**incluido**H2-dftre765-24**LOCAL" with the entity "Espanha". In this case, the relation type has been simplified to "incluido" and the subject and object entity type information, as well as the object entity identifier, has been removed.

The other change was to reduce the number of relation types. After an analysis of the distribution of relations in the dataset, it was observed that some of these relations had few instances and that they could be combined with other similar relations. The relation "nome_de_vinculo_inst", for example, could be combined with the relation "vinculo_inst" without any problem. In cases where the relation had 5 or fewer instances and it was not possible to combine it with other relations, they were discarded. Table 4.5 shows the relations that have been combined or discarded.

After processing, the 48 relation types present in the dataset were reduced to 29 types. The new distribution of relations in the dataset can be seen in Table 4.6.

4.2 NER task

The NER task was performed using the Simple Transformers library. For that, it was necessary to modify the dataset to an input format compatible with the library. In this task, two executions were carried out: one to just identify which words are entities, and the other to classify each one of these entities.

Subsection 4.2.1 provides a brief explanation of the ST library. Subsection 4.2.2 describes the input data format for the model. Finally, subsection 4.2.3 reports how the experiments were carried out.

4.2.1 Simple Transformers library

Simple Transformers (ST) is a Python library developed by Thilina Rajapakse to facilitate the use of Transformer models, which are state-of-the-art NLP systems that use deep learning models and adopt the mechanism of attention (VASWANI et al., 2017). The ST is based on the Transformers library provided by the Hugging Face community.

Each available model has been adapted for a specific NLP task and therefore has some benefits. Among these benefits are the simplified configuration of the model (every model already has a default configuration), no boilerplate code, optimized input data, and clean output data.

The ST model used in this work was related to the NER task. To start using the library, the first step is to choose the class related to the task (in this case, the NERModel class), select the type of supported model (such as BERT, ALBERT, RoBERTa, among others), and configure the general parameters (such as sequence length, for example) and the specific parameters (such as the list of entity classes for the NER).

After choosing the model type, it is possible to specify the exact architecture and trained weights to use through pretrained models. These models may be available directly through Hugging Face or through other community contributors.

With the model properly initialized, it is possible to train and evaluate the model using input data. This can be done by separating the input data into training and test sets with cross-validation, for example. After evaluating the model, the metrics and predictions obtained can be analyzed.

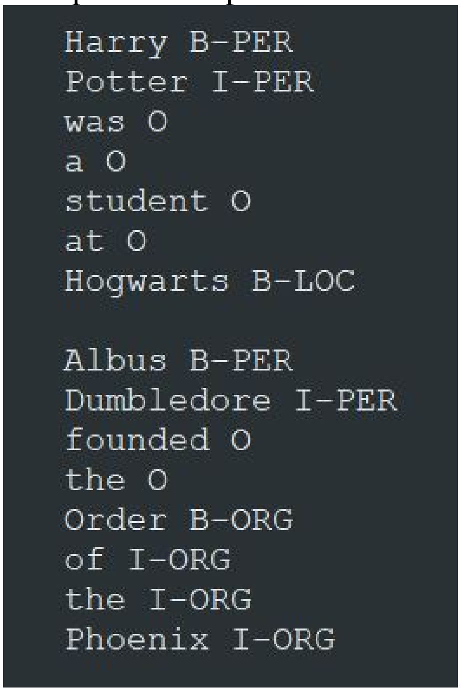
4.2.2 Data format

The input data to the ST NER task can be a path to a text file containing the data. When using text files as input, the data should be in the CoNLL format and tagged with the BIO2 format. The CoNLL format is a text file with one word per line with sentences separated by an empty line. The first column in a line should be the word and the second column should be the label.

BIO2 is a tagging format that uses prefixes to classify labels. When an entity appears, the word is marked with a label that begins with the prefix “B-” followed by the entity class. If the entity is made up of more than one word, the following labels have the prefix “I-”. The words that are not entities are marked with the label “O”.

Figure 4.2 shows an example of what the input data for the model should look like. The first sentence is “Harry Potter was a student at Hogwarts” and has two entities: “Harry Potter”, a person, and “Hogwarts”, a location. The second sentence is “Albus Dumbledore founded the Order of the Phoenix” and needs to be separated from the first sentence by a blank line. In this sentence, there are the entities “Albus Dumbledore” and “Order of the Phoenix”, which have the types of person and organization, respectively.

Figure 4.2: Example of the input format for ST NERModel.



```
Harry B-PER
Potter I-PER
was O
a O
student O
at O
Hogwarts B-LOC

Albus B-PER
Dumbledore I-PER
founded O
the O
Order B-ORG
of I-ORG
the I-ORG
Phoenix I-ORG
```

Source: The Author

4.2.3 Experiments

The experiments were performed using Google Colaboratory (also known as Colab), a Jupyter notebook environment that runs in the cloud. To decrease the task execution time, the environment was configured to use a GPU.

Two sets of inputs in CoNLL format were generated from the dataset. The first version contains only labels “B”, “I”, and “O”, and was used to identify the occurrences of the entities. The second version was used to classify the entities, so the labels that marked the entities contained the class along with the prefix.

The k-fold cross-validation technique was applied to both input versions. The separation of the folds was made by the number of documents in the dataset, that is, as the chosen k value was 10 and the dataset has 129 documents, this means that each fold was composed of 13 documents, except for the last fold that was left with 12. For each of the 10 iterations, a NER model was trained using 9 folds and evaluated using the remaining fold. The result of the evaluation was saved for later analysis.

The model used in both executions was the BERTimbau Base (SOUZA; NOGUEIRA; LOTUFO, 2020), a pretrained BERT model for Brazilian Portuguese developed by NeuralMind. For fine-tuning, it was used only one epoch with a learning rate equal to $4\mathrm{e}{-5}$.

In addition to configuring the labels present in each version, the "max_seq_length" parameter has also been changed. Since the maximum value for this parameter was 512, the model truncated sentences if they exceeded 512 words. The solution found to work around this problem was to split the sentence in two when it became too long. For the rest of the parameters, the defaults of the NERArgs class were maintained[2].

4.3 RE task

The RE task was performed using the Kindred library. Three sets of input data were generated for this task, so three different runs were made.

Subsection 4.3.1 provides a brief explanation of the Kindred library. Subsection 4.3.2 describes the input data format for the model. Finally, subsection 4.3.3 reports how the experiments were carried out.

4.3.1 Kindred library

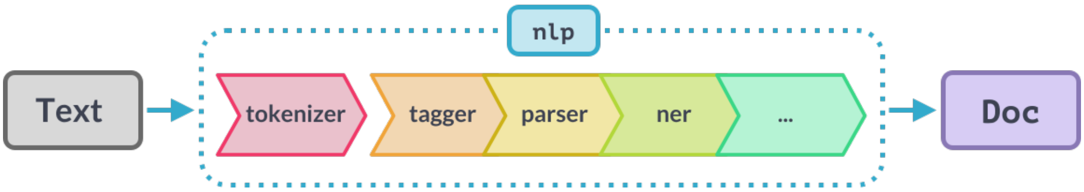
Kindred is a Python library developed by Jake Lever and designed for binary relation extraction from biomedical texts (LEVER; JONES, 2017). It takes a supervised learning approach and therefore requires training data to build a model.

The library uses the spaCy package to perform the parsing. SpaCy is an NLP library that has pretrained pipelines in several languages. Each pipeline receives an input text that will be processed in different components and produce a Doc object. Figure 4.3 shows an example pipeline with 4 components. Therefore, before using Kindred it is necessary to install a pipeline for the corresponding language.

The first step when using the library is to load a Corpus, a collection of text docu-

[2] <www.simpletransformers.ai/docs/usage>

Figure 4.3: Example of a spaCy pipeline with 4 components.



Source: spaCy

ments, from a directory. The directory must contain files in formats supported by Kindred. Tests were performed using two of the supported formats: the BioNLP standoff format and the PubAnnotation JSON format.

With the Corpus properly loaded, the next step is to initialize the RelationClassifier class with any of the models available in spaCy and train the classifier with a Corpus set. The remainder of the Corpus can be used as a test set to evaluate the performance of the model. The library already provides a method that divides the Corpus into k-folds, which facilitates the cross-validation process. After the evaluation, the metrics are obtained for each relation present in the dataset.

4.3.2 Data format

During the preparation of this work, two data formats compatible with Kindred were used. In both cases, the system input was the path to the directory containing the files.

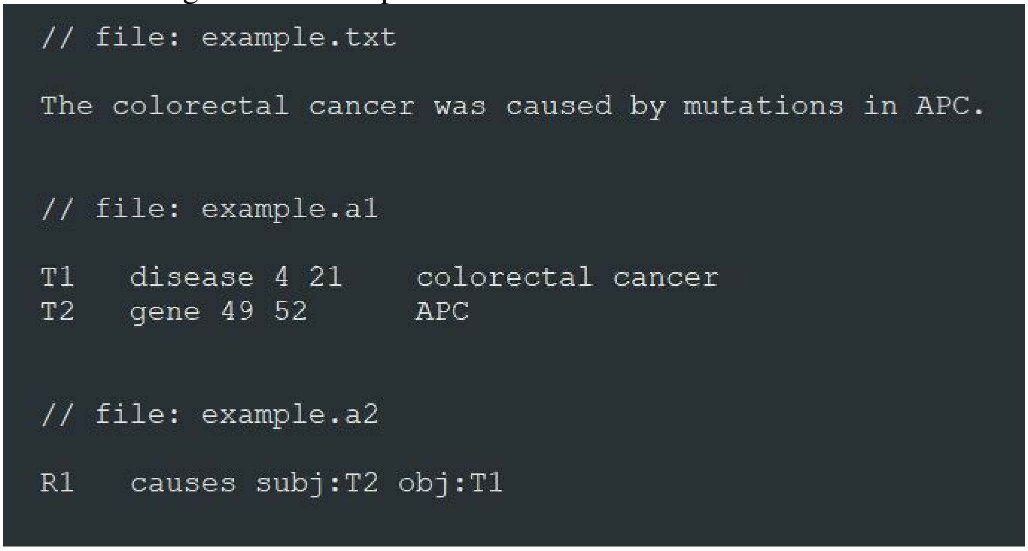
The first tests were carried out using the BioNLP standoff format. A disadvantage of this format is that many files are needed since 3 files are created for each sentence. The first one is a TXT file containing only the sentence.

The second file has the extension ".a1" and contains entity annotations. Each line contains three tab-delimited columns. The first column is the identifier (a T with a number). The second column contains the entity type, start, and end position in the text with spaces in between. And the third column has the entity itself.

The third file has the extension ".a2" and contains the relation annotations. Each line contains two tab-delimited columns. The first column is the identifier (an R with a number). The second column is the relation type and then the arguments of the relation, in the form of "name:entityid". The entity identifier corresponds to the identifier in the ".a1" file.

An example with the 3 files can be seen in Figure 4.4. The TXT file contains the sentence “The colorectal cancer was caused by mutations in APC.”. The “.a1” file describes the two entities present in the sentence: the disease “colorectal cancer”, which appears between positions 4 and 21 of the text, and the gene “APC”, which appears in the range between 49 and 52. Finally, the “.a2” file has the relation “causes” between the subject “APC” and the object “colorectal cancer”.

Figure 4.4: Example of the standoff format for Kindred.



```
// file: example.txt

The colorectal cancer was caused by mutations in APC.


// file: example.a1

T1   disease 4 21    colorectal cancer
T2   gene 49 52      APC


// file: example.a2

R1   causes subj:T2 obj:T1
```

Source: The Author

The second data format used was the PubAnnotation JSON format. This format was the most used during the elaboration of the work because it centralizes all the annotations of entities and relations in a single JSON file per sentence. Figure 4.5 shows an example of what the JSON file format looks like for the same sentence used previously.

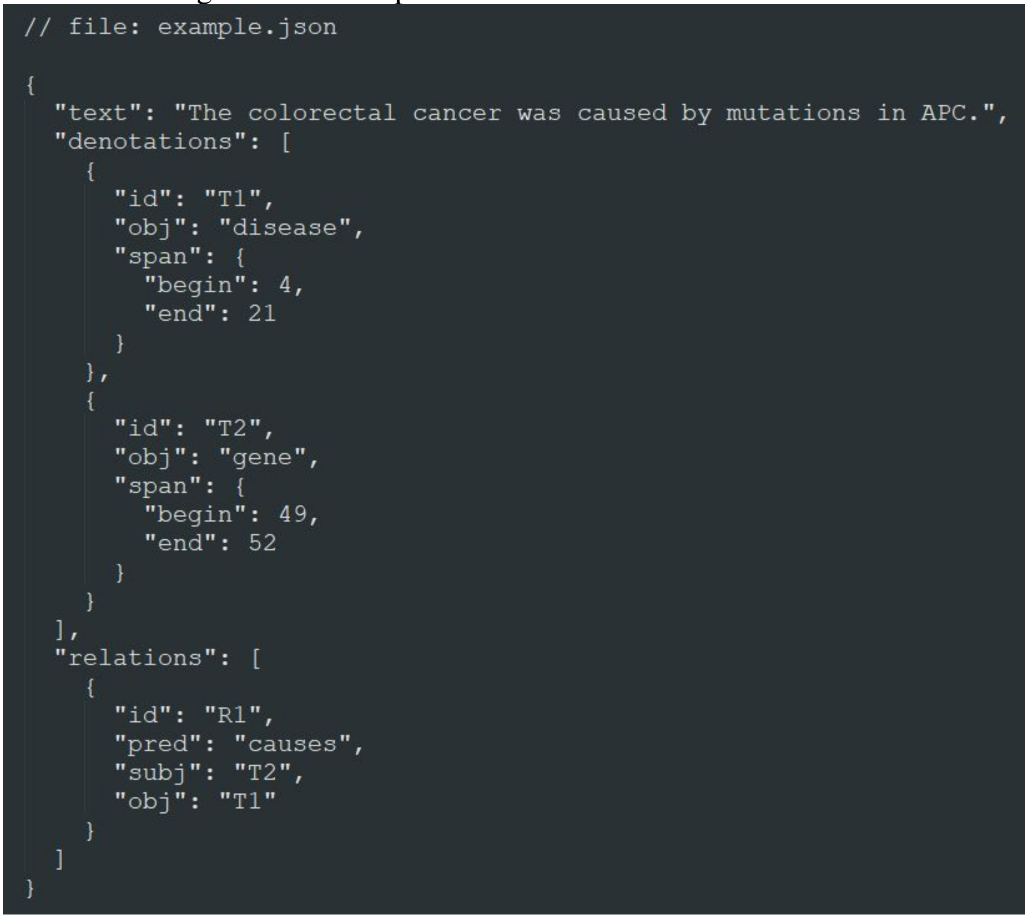
The “text” attribute has the sentence to be analyzed. The “denotations” attribute has a list of all entities present in the sentence. Each entity carries 3 pieces of information: the identifier, its class, and the start and end positions of the entity in the text.

The “relations” attribute has a list of all the relations between the entities. Each relation has 4 attributes: the identifier, the relation type, the entity that acts as the subject, and the entity that acts as the object.

4.3.3 Experiments

Most of the experiments were performed in the Google Colab environment. It was not possible to run one of the experiments in Colab due to the size of the input dataset,

Figure 4.5: Example of the JSON format for Kindred.



```
// file: example.json

{
  "text": "The colorectal cancer was caused by mutations in APC.",
  "denotations": [
    {
      "id": "T1",
      "obj": "disease",
      "span": {
        "begin": 4,
        "end": 21
      }
    },
    {
      "id": "T2",
      "obj": "gene",
      "span": {
        "begin": 49,
        "end": 52
      }
    }
  ],
  "relations": [
    {
      "id": "R1",
      "pred": "causes",
      "subj": "T2",
      "obj": "T1"
    }
  ]
}
```

Source: The Author

which exceeded the memory limit available in the free version of the environment. So this execution was performed locally on a Jupyter notebook.

During the Kindred study, a set of inputs in the standoff format was generated from sentences extracted from DBpedia. The good results obtained in this execution encouraged the use of the library.

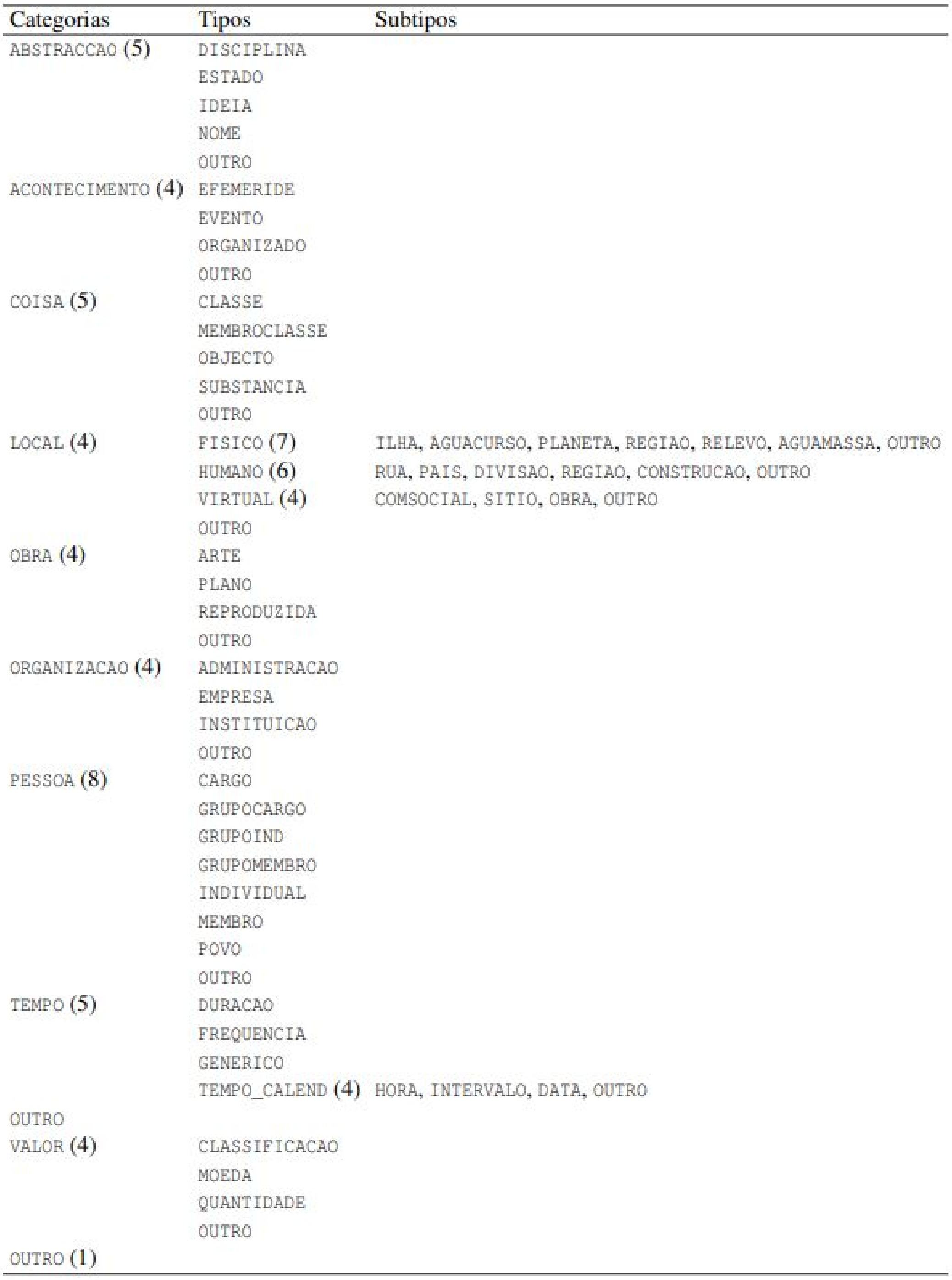
Three input sets in JSON format were generated from the Second HAREM dataset. The first set divided each sentence into a different file with the entities and relations present in that sentence, resulting in 2273 files. A limitation of Kindred's RelationClassifier is that it only identifies relations between entities in the same sentence. Since the dataset had relations between entities of different sentences, these relations had to be discarded. In total, 2914 relations were discarded (about 63% of the relations in the dataset).

To use all relations in the dataset, a second set of JSON files was generated. This time, all the sentences in the same document were concatenated to form a single large sentence. This was done after observing that the entities only had relations with other entities in the same document. Thus, 129 JSON files were created, one for each document in the dataset.

After the execution of the ER in these two input sets, it was decided to create a third set, to improve the metrics obtained. This third set was based on the dataset used in the study stage of the library. The files used in this stage contained only one relation and, if the sentence had more than one relation, another file was generated specifically for that relation. Furthermore, the logic of concatenating the sentences of the same document was maintained. Thus, a set of 4571 JSON files was generated and, possibly due to its size, it was the only one that was not able to run on Colab.

In each of the three input sets, the k-fold cross-validation technique was applied. The number of folds chosen was 10, that is, for each fold to be evaluated, the remaining 9 were used in training. In addition, the spaCy model used in all executions was the "pt_core_news_md", a Portuguese pipeline optimized for CPU created by Explosion. The results of each evaluation will be analyzed in the next chapter.

Table 4.3: Categories, Types and Subtypes of the entities present in the Second HAREM.



Categorias	Tipos	Subtipos
ABSTRACCAO (5)	DISCIPLINA	
	ESTADO	
	IDEIA	
	NOME	
	OUTRO	
ACONTECIMENTO (4)	EFEMERIDE	
	EVENTO	
	ORGANIZADO	
	OUTRO	
COISA (5)	CLASSE	
	MEMBROCLASSE	
	OBJECTO	
	SUBSTANCIA	
	OUTRO	
LOCAL (4)	FISICO (7)	ILHA, AGUACURSO, PLANETA, REGIAO, RELEVO, AGUAMASSA, OUTRO
	HUMANO (6)	RUA, PAIS, DIVISAO, REGIAO, CONSTRUCAO, OUTRO
	VIRTUAL (4)	COMSOCIAL, SITIO, OBRA, OUTRO
	OUTRO	
OBRA (4)	ARTE	
	PLANO	
	REPRODUZIDA	
	OUTRO	
ORGANIZACAO (4)	ADMINISTRACAO	
	EMPRESA	
	INSTITUICAO	
	OUTRO	
PESSOA (8)	CARGO	
	GRUPOCARGO	
	GRUPOIND	
	GRUPOMEMBRO	
	INDIVIDUAL	
	MEMBRO	
	POVO	
	OUTRO	
TEMPO (5)	DURACAO	
	FREQUENCIA	
	GENERICO	
	TEMPO_CALEND (4)	HORA, INTERVALO, DATA, OUTRO
OUTRO		
VALOR (4)	CLASSIFICACAO	
	MOEDA	
	QUANTIDADE	
	OUTRO	
OUTRO (1)		

Source: (MOTA; SANTOS, 2008)

Table 4.4: New distribution of entities in the dataset and mapping rules used.

Entity type	*#*	*Rules*
ABSTRACCAO	292	CATEG=ABSTRACCAO
ACONTECIMENTO	323	CATEG=ACONTECIMENTO

INDIVIDUO	1774	CATEG=PESSOA & TIPO=INDIVIDUAL
		CATEG=PESSOA & TIPO=CARGO

LOCAL	1539	CATEG=LOCAL
OBRA	489	CATEG=OBRA

ORGANIZACAO	1459	CATEG=ORGANIZACAO
		CATEG=PESSOA & TIPO=GRUPOCARGO
		CATEG=PESSOA & TIPO=GRUPOIND
		CATEG=PESSOA & TIPO=GRUPOMEMBRO
		CATEG=PESSOA & TIPO=POVO

OUTRO	390	CATEG=OUTRO
		CATEG=COISA

TEMPO	1199	CATEG=TEMPO
VALOR	352	CATEG=VALOR
TOTAL	7817	

Source: The Author

Table 4.5: Relation types that have been combined or discarded.

Relation type	*Reason*
consequencia_de	Discarded because it only had 1 instance
local_morte	Discarded because it only had 4 instances
local_nascimento_de	Merged with natural_de
localizacao_de	Discarded because it only had 5 instance
nome_de	Merged with ident
nome_de_data_de	Merged with datado_de
nome_de_ident	Merged with ident
nome_de_inclui	Merged with inclui
nome_de_incluido	Merged with incluido
nome_de_obra_de	Merged with obra_de
nome_de_outrarel	Merged with outrarel
nome_de_pratica_se	Merged with pratica_se
nome_de_praticado_por	Merged with praticado_por
nome_de_vinculo_inst	Merged with vinculo_inst
nomeado_por	Discarded because it only had 5 instances
ocorre_em	Merged with localizado_em
outra_edicao	Discarded because it only had 3 instances
periodo_vida	Discarded because it only had 5 instances
residente_de	Discarded because it only had 3 instances

Source: The Author

Table 4.6: New distribution of relations in the dataset.

Relation type	*#*
autor_de	54
causador_de	22
data_de	76
data_morte	9
data_nascimento	6
datado_de	10
ident	2201
inclui	320
incluido	507
localizado_em	121
natural_de	133
obra_de	93
outrarel	97
participante_em	90
personagem_de	14
pratica_se	16
praticado_em	42
praticado_por	16
praticante_de	26
produtor_de	28
produzido_por	22
propriedade_de	18
proprietario_de	20
relacao_familiar	82
relacao_profissional	17
residencia_de	15
sede_de	196
ter_participacao_de	64
vinculo_inst	256
TOTAL	4571

Source: The Author

5 RESULTS AND ANALYSIS

This chapter presents the results obtained in the experiments, as well as the analysis of these results. Section 5.1 presents the results obtained in the NER task, while section 5.2 presents the results of the RE task.

5.1 NER task

As already mentioned, the NER task used the Simple Transformers library in all executions. One of the available outputs from the library's NERModel was a list of all the labels that were predicted from the input dataset. These predictions were compared directly with the input labels using the scikit-learn library, a Python ML library that already has methods for extracting metrics. The execution time with all folds of the cross-validation was about 25 minutes, both in the task of identification and classification of entities.

The first results analyzed were from the task of identifying entities, that is, the labels used were only "B", "I", and "O", without the class information. Table 5.1 presents the precision, recall, and F-score metrics, both at MACRO-level and MICRO-level, obtained when performing entity identification. It is important to note that, in the case of multiclass classification, the MICRO metrics are always the same, since, for each false positive, there will always be a false negative and vice versa. The results obtained in this task were very encouraging.

Table 5.1: MACRO and MICRO metrics from the entity identification task.

		MACRO			MICRO
Fold	*Total words*	*Precision*	*Recall*	*F-score*	*P=R=F*
0	9027	95.49%	94.93%	95.20%	98.19%
1	10439	95.12%	95.14%	95.13%	97.27%
2	8695	95.55%	96.74%	96.12%	98.14%
3	3415	94.82%	94.14%	94.48%	96.89%
4	3350	97.48%	96.59%	**97.03%**	98.50%
5	5888	94.52%	95.47%	94.99%	98.01%
6	6382	96.65%	**96.98%**	96.80%	98.43%
7	8238	92.96%	95.38%	94.14%	97.83%
8	16241	**97.54%**	96.51%	97.01%	**98.52%**
9	5755	93.35%	96.05%	94.64%	96.49%
	Average	95.35%	95.79%	95.55%	97.83%

Source: The Author

The next step was to analyze the results of the entity classification task. In this case, the labels of the input files contained the entity class in combination with some BIO prefix. The metrics obtained in this execution can be seen in Table 5.2.

Table 5.2: MACRO and MICRO metrics from the entity classification task.

		MACRO			MICRO
Fold	*Total words*	*Precision*	*Recall*	*F-score*	*P=R=F*
0	9027	60.27%	60.21%	58.41%	95.31%
1	10439	62.07%	60.57%	60.27%	93.18%
2	8695	61.81%	61.35%	60.52%	95.13%
3	3415	**68.85%**	68.11%	**67.52%**	94.41%
4	3350	62.98%	58.42%	57.98%	93.49%
5	5888	67.25%	**69.64%**	66.05%	**96.38%**
6	6382	63.96%	63.95%	62.51%	94.31%
7	8238	65.03%	63.36%	62.87%	95.26%
8	16241	63.58%	55.92%	53.05%	92.75%
9	5755	66.29%	63.76%	62.04%	91.21%
	Average	64,21%	62,53%	61,12%	94,14%

Source: The Author

Two observations can be made from these values. The first one is related to the discrepancy of MACRO metrics between classification and identification tasks. The F-score, for example, had a difference of approximately 34% between the two tasks. This difference shows that classifying entities is much more complex than just identifying them.

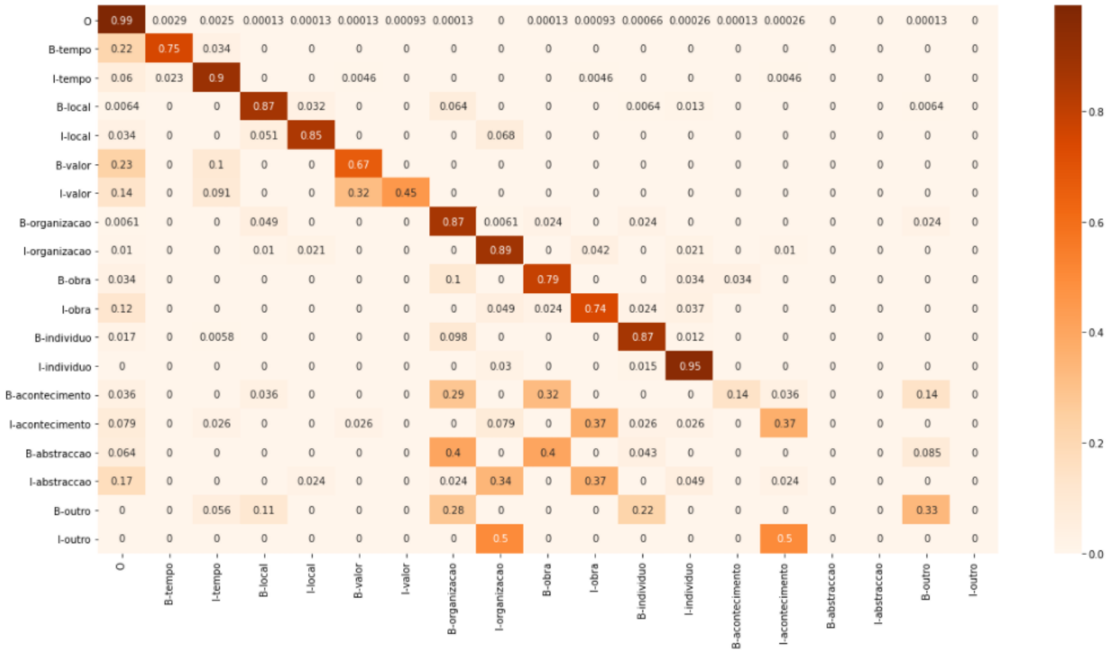
The second observation refers to the large difference between the MACRO and MICRO metrics (about 30%). This is because the model presents a better performance for certain classes of entities. Figures 5.3 and 5.4 show the normalized confusion matrices for fold 0 and fold 5 created during cross-validation, respectively. All values in the same row of the matrix refer to the ground truth and all values in the same column are the predictions.

When analyzing the matrices, it is possible to observe that the classes “individuo” and “organização” showed good results. Meanwhile, the model has rarely been able to predict entities of type “abstraccao” and “outro”. One of the reasons for this disparity in the results is related to the number of instances of each class. While the classes “individuo” and “organização” have about 1500 instances, the classes “abstraccao” and “outro” have 292 and 390 instances, respectively.

Another factor is related to the meanings of the entities. While individuals and organizations represent concrete entities, such as “Bill Gates” or “Microsoft”, the entities “other” and “abstraction” present more abstract concepts, such as “Portuguese Language” or “Minimum Wage”.

Moreover, since the class “other” includes all entities that could not be classified in the other classes, it ends up becoming complex with a variety of entities. These characteristics end up influencing the performance of the system since the model has more difficulty in detecting patterns in these classes. Another important detail is that the predictions for the label “O” showed a high hit rate in all folds, which widens this difference between the MACRO and MICRO metrics.

Table 5.3: Confusion matrix of fold 0 of the entity classification task.

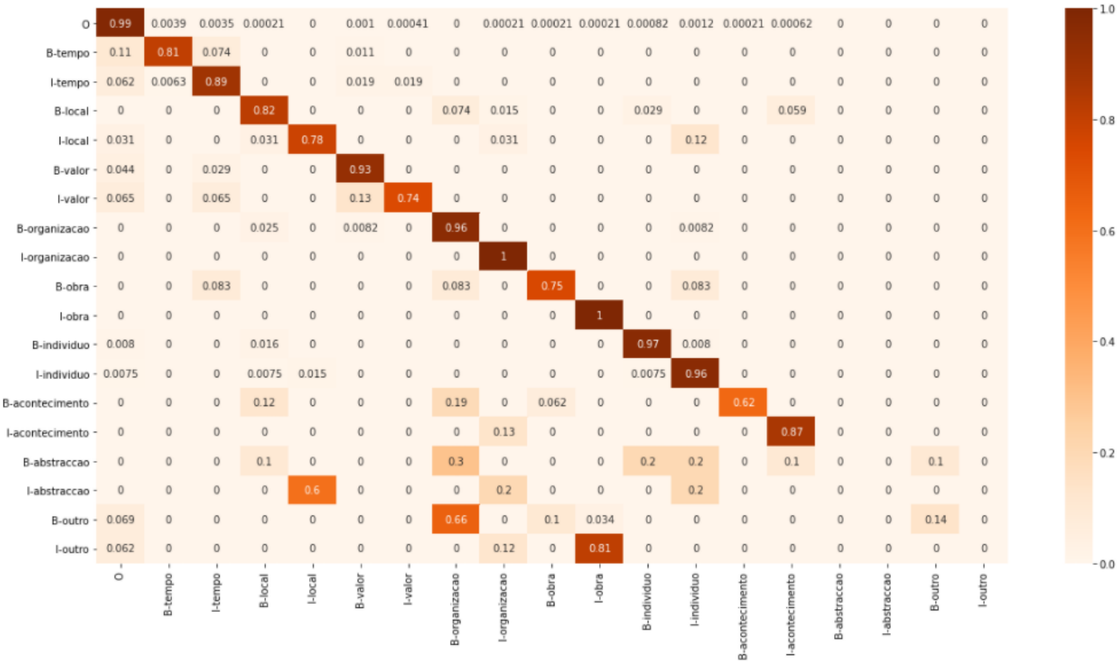


	O	B-tempo	I-tempo	B-local	I-local	B-valor	I-valor	B-organizacao	I-organizacao	B-obra	I-obra	B-individuo	I-individuo	B-acontecimento	I-acontecimento	B-abstraccao	I-abstraccao	B-outro	I-outro
O	0.99	0.0029	0.0025	0.00013	0.00013	0.00013	0.00093	0.00013	0	0.00013	0.00093	0.00066	0.00026	0.00013	0.00026	0	0	0.00013	0
B-tempo	0.22	0.75	0.034	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
I-tempo	0.06	0.023	0.9	0	0	0.0046	0	0	0	0	0.0046	0	0	0	0.0046	0	0	0	0
B-local	0.0064	0	0	0.87	0.032	0	0	0.064	0	0	0	0.0064	0.013	0	0	0	0	0.0064	0
I-local	0.034	0	0	0.051	0.85	0	0	0	0.068	0	0	0	0	0	0	0	0	0	0
B-valor	0.23	0	0.1	0	0	0.67	0	0	0	0	0	0	0	0	0	0	0	0	0
I-valor	0.14	0	0.091	0	0	0.32	0.45	0	0	0	0	0	0	0	0	0	0	0	0
B-organizacao	0.0061	0	0	0.049	0	0	0	0.87	0.0061	0.024	0	0.024	0	0	0	0	0	0.024	0
I-organizacao	0.01	0	0	0.01	0.021	0	0	0	0.89	0	0.042	0	0.021	0	0.01	0	0	0	0
B-obra	0.034	0	0	0	0	0	0	0.1	0	0.79	0	0	0.034	0.034	0	0	0	0	0
I-obra	0.12	0	0	0	0	0	0	0	0.049	0.024	0.74	0.024	0.037	0	0	0	0	0	0
B-individuo	0.017	0	0.0058	0	0	0	0	0.098	0	0	0	0.87	0.012	0	0	0	0	0	0
I-individuo	0	0	0	0	0	0	0	0	0.03	0	0	0.015	0.95	0	0	0	0	0	0
B-acontecimento	0.036	0	0	0.036	0	0	0	0.29	0	0.32	0	0	0	0.14	0.036	0	0	0.14	0
I-acontecimento	0.079	0	0.026	0	0	0.026	0	0	0.079	0	0.37	0.026	0.026	0	0.37	0	0	0	0
B-abstraccao	0.064	0	0	0	0	0	0	0.4	0	0.4	0	0.043	0	0	0	0	0	0.085	0
I-abstraccao	0.17	0	0	0	0.024	0	0	0.024	0.34	0	0.37	0	0.049	0	0.024	0	0	0	0
B-outro	0	0	0.056	0.11	0	0	0	0.28	0	0	0	0.22	0	0	0	0	0	0.33	0
I-outro	0	0	0	0	0	0	0	0	0.5	0	0	0	0	0	0.5	0	0	0	0

Source: The Author

After comparing the value of the MACRO F-score with the values of the other systems studied, the obtained F-score was slightly better. While systems like Priberam and CRF+LG have an F-score of 57.11% and 57.8%, respectively, the model proposed in this work presents an F-score of 61.12%. However, these values cannot be compared directly, as the datasets are not equivalent. Several changes were made to the dataset during the course of this work, and even if there were no such changes, the original dataset itself was already different. The Segundo HAREM GC used in this work was an updated version of the collection used in the Workshop.

Table 5.4: Confusion matrix of fold 5 of the entity classification task.



	O	B-tempo	I-tempo	B-local	I-local	B-valor	I-valor	B-organizacao	I-organizacao	B-obra	I-obra	B-individuo	I-individuo	B-acontecimento	I-acontecimento	B-abstraccao	I-abstraccao	B-outro	I-outro
O	0.99	0.0039	0.0035	0.00021	0	0.001	0.00041	0	0.00021	0.00021	0.00021	0.00082	0.0012	0.00021	0.00062	0	0	0	0
B-tempo	0.11	0.81	0.074	0	0	0.011	0	0	0	0	0	0	0	0	0	0	0	0	0
I-tempo	0.062	0.0063	0.89	0	0	0.019	0.019	0	0	0	0	0	0	0	0	0	0	0	0
B-local	0	0	0	0.82	0	0	0	0.074	0.015	0	0	0.029	0	0	0.059	0	0	0	0
I-local	0.031	0	0	0.031	0.78	0	0	0	0.031	0	0	0	0.12	0	0	0	0	0	0
B-valor	0.044	0	0.029	0	0	0.93	0	0	0	0	0	0	0	0	0	0	0	0	0
I-valor	0.065	0	0.065	0	0	0.13	0.74	0	0	0	0	0	0	0	0	0	0	0	0
B-organizacao	0	0	0	0.025	0	0.0082	0	0.96	0	0	0	0	0.0082	0	0	0	0	0	0
I-organizacao	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
B-obra	0	0	0.083	0	0	0	0	0.083	0	0.75	0	0	0.083	0	0	0	0	0	0
I-obra	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
B-individuo	0.008	0	0	0.016	0	0	0	0	0	0	0	0.97	0.008	0	0	0	0	0	0
I-individuo	0.0075	0	0	0.0075	0.015	0	0	0	0	0	0	0.0075	0.96	0	0	0	0	0	0
B-acontecimento	0	0	0	0.12	0	0	0	0.19	0	0.062	0	0	0	0.62	0	0	0	0	0
I-acontecimento	0	0	0	0	0	0	0	0	0.13	0	0	0	0	0	0.87	0	0	0	0
B-abstraccao	0	0	0	0.1	0	0	0	0.3	0	0	0	0.2	0.2	0	0.1	0	0	0.1	0
I-abstraccao	0	0	0	0	0.6	0	0	0	0.2	0	0	0	0.2	0	0	0	0	0	0
B-outro	0.069	0	0	0	0	0	0	0.66	0	0.1	0.034	0	0	0	0	0	0	0.14	0
I-outro	0.062	0	0	0	0	0	0	0	0.12	0	0.81	0	0	0	0	0	0	0	0

Source: The Author

5.2 RE task

Unlike Simple Transformers, the Kindred library already presents the metrics directly in the system output. In addition to precision, recall, and f-score, Kindred also displays the number of TP, FP, and FN for each of the relations.

To facilitate reading, the results of the experiments were separated into three runs, according to the input dataset. Run 1 used the set in which each HAREM sentence had a corresponding JSON file. Run 2 used the set in which each HAREM document had a corresponding JSON file. Finally, Run 3 used the set in which each HAREM relation had a corresponding JSON file.

The results of Run 1 can be seen in Table 5.5. This run took about 40 minutes. Since the calculated metrics were not very good, both at MACRO-level and MICRO-level, another dataset was generated to perform another run. One hypothesis for the poor performance of Run 1 is that many relations needed to be discarded during the creation of the input set, so the purpose of Run 2 is to take advantage of all relations present in the dataset.

Table 5.6 shows the results of Run 2. This run was done with 129 input files divided into 8 folds, unlike the previous run that used 1905 files in 10 folds. Moreover, the execution time was shorter: just 25 minutes. From what was observed, all metrics of

Table 5.5: MACRO and MICRO metrics obtained in Run 1 of the RE task.

	MACRO			MICRO		
Fold	*P*	*R*	*F*	*P*	*R*	*F*
0	14.99%	4.34%	5.66%	21.54 %	7.37 %	10.98%
1	8.31%	6.12%	6.70%	20.69%	6.12%	9.45%
2	15.56%	8.12%	10.23%	**22.37%**	12.98%	**16.42%**
3	14.11%	**11.97%**	**12.04%**	19.40%	9.56%	12.81%
4	6.82%	6.07%	5.49%	11.11%	6.67%	8.33%
5	6.71%	6.24%	6.27%	14.89%	9.46%	11.57%
6	**16.98%**	6.13%	8.48%	11.90%	8.55%	9.95%
7	9.76%	7.33%	7.85%	18.42%	6.67%	9.79%
8	15.28%	4.50%	6.47%	15.85%	5.94%	8.64%
9	11.92%	8.14%	9.45%	18.95%	**13.43%**	15.72%
Avg.	12,05%	6,89%	7,86%	17,51%	8,67%	11,37%

Source: The Author

Run 2 were worse than those of Run 1. Since this execution used 100% of the dataset relations, a better result was expected, but this did not happen. This probably happened because the text to be processed in each JSON was much larger than the texts in Run 1 and this ended up hampering the RE process. As the number of TPs remained the same as the previous run, the metrics ended up being lower, because the number of relations was much higher (that is, more FPs and FNs).

Table 5.6: MACRO and MICRO metrics obtained in Run 2 of the RE task.

	MACRO			MICRO		
Fold	*P*	*R*	*F*	*P*	*R*	*F*
0	9.08%	2.81%	3.69%	12.77%	2.43%	4.08%
1	11.29%	2.32%	3.58%	12.18%	2.57%	4.24%
2	3.79%	1.29%	1.79%	10.34%	1.58%	2.75%
3	9.53%	**4.37%**	**5.51%**	15.79%	3.28%	5.43%
4	7.49%	3.11%	4.14%	**25.00%**	**4.37%**	**7.44%**
5	5.95%	4.27%	4.77%	17.09%	3.08%	5.21%
6	**11.44%**	3.56%	4.79%	**25.00%**	3.72%	6.48%
7	3.67%	2.75%	2.59%	11.21%	2.56%	4.17%
Avg.	7,78%	3,06%	3,86%	16,17%	2,95%	4,97%

Source: The Author

Since the metrics of the other runs were very low, another approach was used to try to get better results. During the study of the Kindred library, the input set used contained only one relation per file, unlike the set of Run 2. With this input set, extracted from DBPedia, the model reached an average MICRO F-score of 85.47% and an average MACRO F-score of 62.73%, which encouraged the use of the library.

Therefore, Run 3 seeks to replicate this behavior by maintaining only one relation for each file. The results of Run 3 can be seen in Table 5.7. Since the size of the input set was relatively large, with 4570 JSON files, this was the only run that needed to be done locally, because it exceeded the available memory in Colab. These factors also affected the execution time, which reached approximately 2 hours. Even so, it is easy to see the advantages of this approach, since all the metrics in the system were better.

It is difficult to analyze these differences between Runs 2 and 3 since both input sets have the same number of sentences, entities, and relations. The only difference between them was the number of relations that were allocated to each file. This is possibly due to some internal implementation detail of Kindred, which ends up causing this type of behavior.

Table 5.7: MACRO and MICRO metrics obtained in Run 3 of the RE task.

	MACRO			MICRO		
Fold	*P*	*R*	*F*	*P*	*R*	*F*
0	42.28%	**18.46%**	**23.47%**	56.31%	12.69%	20.71%
1	33.76%	15.56%	19.24%	51.16%	9.63%	16.21%
2	40.82%	17.13%	22.99%	52.13%	10.72%	17.78%
3	33.42%	15.62%	19.62%	51.96%	11.59%	18.96%
4	25.10%	10.36%	13.73%	45.20%	7.22%	12.45%
5	34.89%	14.18%	18.47%	58.23%	10.06%	17.16%
6	34.72%	17.85%	21.82%	55.14%	**12.91%**	**20.92%**
7	35.92%	14.31%	19.16%	52.22%	10.28%	17.18%
8	**43.75%**	16.18%	22.17%	**58.89%**	11.59%	19.38%
9	38.85%	15.15%	20.16%	53.85%	12.25%	19.96%
Avg.	36,35%	15,48%	20,08%	53,51%	10,89%	18,07%

Source: The Author

Even so, the performance of the system was inferior to the other related works. The REMBRANDT and SeRELeP systems, which participated in the RE task at the Second HAREM Workshop, achieved a MACRO F-score of 45.02% and 36.65%, respectively. The RelP system, which did not participate in the Workshop but also used the HAREM datasets, also obtained a higher F-score, around 43%.

Since the dataset used in this work is not the same as the one used in the other works mentioned, it is not possible to directly compare the results. Even so, it is possible to get an idea of the performance of the system.

One possible explanation is that the Kindred library works better with smaller sentences instead of the long sentences used in Run 3. However, when using smaller sentences, several relations were eventually discarded (as in Run 1).

Another important detail is that it was not possible to infer some relations directly from the texts, as external knowledge was necessary for this. An example of this occurs between the entity “Alemanha” and the entity “Europa”. These two entities have a relation of type “incluido” since Germany is part of Europe, however, this information is not explicit in any part of the text.

This problem occurs mainly with the relations of identity and inclusion. The identity relation indicated the equivalence between two entities, that is, it signaled that two entities in the text were, in fact, the same entity. This can be seen in Table 5.8, which shows the distribution of TP, FP, and FN predictions for folds 5 and 6 of Run 3.

Another detail about this distribution is that there are many more occurrences of FN than of FP. This indicates that the system problem is more related to the fact that it is unable to identify the relations than to predicting the incorrect type.

One of the goals of this work was to use the entities and relations extracted from the Second HAREM to build a KG. However, the performance of the NER and RE models ended up being below expectations. Furthermore, if the RE task were applied directly to entities obtained after the NER, the problem would become even more complex. Therefore, the construction of the KG will be postponed for future work.

Table 5.8: Distribution of predictions for folds 5 and 6 in Run 3 of the RE task.

	Fold 5			Fold 6		
Relation type	*TP*	*FP*	*FN*	*TP*	*FP*	*FN*
autor_de	4	0	4	3	1	1
causador_de	0	0	1	0	0	2
data_de	6	0	4	6	0	0
data_morte	0	0	2	0	0	1
data_nascimento	0	0	0	0	1	0
datado_de	1	0	0	0	0	1
ident	**8**	**5**	**203**	**12**	**12**	**210**
inclui	**7**	**11**	**30**	**5**	**8**	**20**
incluido	**3**	**3**	**50**	**10**	**8**	**51**
localizado_em	1	2	12	0	2	15
natural_de	0	0	18	1	4	15
obra_de	0	2	4	2	0	6
outrarel	3	2	7	3	0	3
participante_em	1	0	8	1	0	6
personagem_de	0	0	3	1	0	2
pratica_se	0	0	1	0	0	2
praticado_em	1	0	3	2	1	2
praticado_por	0	0	3	0	1	1
praticante_de	0	1	1	0	0	2
produtor_de	0	0	5	0	1	2
produzido_por	0	0	1	0	0	2
propriedade_de	0	0	3	0	0	3
proprietario_de	0	0	2	0	0	4
relacao_familiar	3	3	8	0	0	6
relacao_profissional	0	0	1	0	0	4
residencia_de	0	0	2	0	0	1
sede_de	2	1	19	7	5	13
ter_participacao_de	1	2	7	3	0	1
vinculo_inst	5	1	9	3	4	22
TOTAL	46	33	411	59	48	398

Source: The Author

6 CONCLUSION

Named Entity Recognition and Relation Extraction tasks are essential in the process of extracting information from texts. This information can be in the form of tuples, with each tuple containing a relation between entities present in the texts. From this, it is possible to build a database, such as a Knowledge Graph, which is a set of tuples, and which can be used in applications such as Question Answering systems.

However, most of the NER and RE models found in the literature are focused on the English language. Moreover, many Portuguese models adopt a rule-based approach instead of taking advantage of machine learning models. Therefore, the purpose of this work was the elaboration of a system capable of extracting entities and relations in Portuguese using machine learning.

The first step of this work was to study the concepts and related works. Afterward, research was made about which programming language would be used during development, mainly taking into account the available libraries, and which dataset would be chosen to feed the system. Finally, an analysis of the results obtained in the experiments was made.

The system was trained and evaluated using a version of the Second HAREM Golden Collection dataset with a reduced number of classes. After analyzing the results, it was observed that the metrics obtained were below the metrics found in the literature. Since the dataset used in this work was not the same as the one used in related works, it is not possible to make a direct comparison with the results obtained in the literature. Therefore, it would be interesting to adapt the works for the same dataset in the future.

Based on the results of this work, it was concluded that the extraction method was not the most appropriate since the results did not reach expectations, especially the results obtained in the RE task. Because of this, the automated construction of KG was not carried out, since the tuples that would be generated by the system would not be sufficiently reliable for that KG to be used safely.

For future work, it would be interesting to develop new models to obtain better performances, mainly in the RE task. This can be achieved through a more in-depth study of the parameters present in the Simple Transformers and Kindred libraries. Another possibility would be to use other NLP libraries, such as spaCy and Transformers. Both libraries are more complex than those used in this work, but they appear to be more effective for these tasks.

REFERENCES

ABREU, S. C. de; BONAMIGO, T. L.; VIEIRA, R. A review on relation extraction with an eye on portuguese. **Journal of the Brazilian Computer Society**, Springer, v. 19, n. 4, p. 553–571, 2013.

ABREU, S. C. de; VIEIRA, R. Relp: Portuguese open relation extraction. **KO KNOWLEDGE ORGANIZATION**, Nomos Verlagsgesellschaft mbH & Co. KG, v. 44, n. 3, p. 163–177, 2017.

AMARAL, C. et al. Adaptação do sistema de reconhecimento de entidades mencionadas da priberam ao harem. **Desafios na avaliação conjunta do reconhecimento de entidades mencionadas: O Segundo HAREM. Linguateca**, 2008.

AMARAL, D. O. F. do; VIEIRA, R. Nerp-crf: uma ferramenta para o reconhecimento de entidades nomeadas por meio de conditional random fields. **Linguamática**, v. 6, n. 1, p. 41–49, 2014.

BERRAR, D. Cross-validation. **Encyclopedia of bioinformatics and computational biology**, Academic, v. 1, p. 542–545, 2019.

BRUCKSCHEN, M. et al. Sistema serelep para o reconhecimento de relaç oes entre entidades mencionadas. **Mota and Santos (Mota and Santos, 2008)**, 2008.

CARDOSO, N. Rembrandt-reconhecimento de entidades mencionadas baseado em relaçoes e análise detalhada do texto. **quot; Encontro do Segundo HAREM (Universidade de Aveiro Portugal 7 de Setembro de 2008)**, 2008.

CASTRO, P. V. Q. de; SILVA, N. F. F. da; SOARES, A. da S. Portuguese named entity recognition using lstm-crf. In: SPRINGER. **International Conference on Computational Processing of the Portuguese Language**. [S.l.], 2018. p. 83–92.

CHAVES, M. Geo-ontologias e padrões para reconhecimento de locais e de suas relações em textos: o sei-geo no segundo harem. **quot; In Cristina Mota; Diana Santos (ed) Desafios na avaliação conjunta do reconhecimento de entidades mencionadas: O Segundo HAREM Linguateca 2008**, Linguateca, 2008.

CHOWDHURY, G. G. Natural language processing. **Annual review of information science and technology**, Wiley Online Library, v. 37, n. 1, p. 51–89, 2003.

DALIANIS, H. Evaluation metrics and evaluation. In: **Clinical Text Mining**. [S.l.]: Springer, 2018. p. 45–53.

ETZIONI, O. et al. Open information extraction from the web. **Communications of the ACM**, ACM New York, NY, USA, v. 51, n. 12, p. 68–74, 2008.

FREITAS, C. et al. Second harem: advancing the state of the art of named entity recognition in portuguese. In: EUROPEAN LANGUAGE RESOURCES ASSOCIATION. **quot; In Nicoletta Calzolari; Khalid Choukri; Bente Maegaard; Joseph Mariani; Jan Odijk; Stelios Piperidis; Mike Rosner; Daniel Tapias (ed) Proceedings of the International Conference on Language Resources and Evaluation (LREC 2010)(Valletta 17-23 May de 2010) European Language Resources Association**. [S.l.], 2010.

FREITAS, C. et al. Detection of relations between named entities: report of a shared task. In: **quot; In Proceedings of the NAACL HLT Workshop on Semantic Evaluations: Recent Achievements and Future Directions (Boulder June 4)**. [S.l.: s.n.], 2009.

GRANDINI, M.; BAGLI, E.; VISANI, G. Metrics for multi-class classification: an overview. **arXiv preprint arXiv:2008.05756**, 2020.

HOFFART, J. et al. Robust disambiguation of named entities in text. In: **Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing**. [S.l.: s.n.], 2011. p. 782–792.

HOSSIN, M.; SULAIMAN, M. A review on evaluation metrics for data classification evaluations. **International Journal of Data Mining & Knowledge Management Process**, Academy & Industry Research Collaboration Center (AIRCC), v. 5, n. 2, p. 1, 2015.

HUANG, Y. Y.; WANG, W. Y. Deep residual learning for weakly-supervised relation extraction. **arXiv preprint arXiv:1707.08866**, 2017.

HULST, J. M. van et al. Rel: An entity linker standing on the shoulders of giants. **arXiv preprint arXiv:2006.01969**, 2020.

JAMES, G. et al. **An introduction to statistical learning**. [S.l.]: Springer, 2013.

JIANG, J. Information extraction from text. In: **Mining text data**. [S.l.]: Springer, 2012. p. 11–41.

KOLITSAS, N.; GANEA, O.-E.; HOFMANN, T. End-to-end neural entity linking. **arXiv preprint arXiv:1808.07699**, 2018.

LEVER, J.; JONES, S. J. Painless Relation Extraction with Kindred. **BioNLP 2017**, p. 176, 2017.

LIDDY, E. D. Natural language processing. 2001.

LIU, Q. et al. Hierarchical random walk inference in knowledge graphs. In: **Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval**. [S.l.: s.n.], 2016. p. 445–454.

MOTA, C.; SANTOS, D. **Desafios na avaliação conjunta do reconhecimento de entidades mencionadas: O Segundo HAREM**. [S.l.]: Linguateca, 2008.

MUHAMMAD, I. et al. Open information extraction for knowledge graph construction. In: SPRINGER. **International Conference on Database and Expert Systems Applications**. [S.l.], 2020. p. 103–113.

NADEAU, D.; SEKINE, S. A survey of named entity recognition and classification. **Lingvisticae Investigationes**, John Benjamins, v. 30, n. 1, p. 3–26, 2007.

NICKEL, M. et al. A review of relational machine learning for knowledge graphs. **Proceedings of the IEEE**, IEEE, v. 104, n. 1, p. 11–33, 2015.

NIKLAUS, C. et al. A survey on open information extraction. **arXiv preprint arXiv:1806.05599**, 2018.

PICCINNO, F.; FERRAGINA, P. From tagme to wat: a new entity annotator. In: **Proceedings of the first international workshop on Entity recognition & disambiguation**. [S.l.: s.n.], 2014. p. 55–62.

PIROVANI, J.; OLIVEIRA, E. Portuguese named entity recognition using conditional random fields and local grammars. In: **Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)**. [S.l.: s.n.], 2018.

RAIMAN, J.; RAIMAN, O. Deeptype: multilingual entity linking by neural type system evolution. In: **Proceedings of the AAAI Conference on Artificial Intelligence**. [S.l.: s.n.], 2018. v. 32, n. 1.

REFAEILZADEH, P.; TANG, L.; LIU, H. Cross-validation. **Encyclopedia of database systems**, Springer: New York, v. 5, p. 532–538, 2009.

SARAWAGI, S. **Information extraction**. [S.l.]: Now Publishers Inc, 2008.

SHEN, W.; WANG, J.; HAN, J. Entity linking with a knowledge base: Issues, techniques, and solutions. **IEEE Transactions on Knowledge and Data Engineering**, IEEE, v. 27, n. 2, p. 443–460, 2014.

SOUZA, F.; NOGUEIRA, R.; LOTUFO, R. Portuguese named entity recognition using bert-crf. **arXiv preprint arXiv:1909.10649**, 2019.

SOUZA, F.; NOGUEIRA, R.; LOTUFO, R. BERTimbau: pretrained BERT models for Brazilian Portuguese. In: **9th Brazilian Conference on Intelligent Systems, BRACIS, Rio Grande do Sul, Brazil, October 20-23 (to appear)**. [S.l.: s.n.], 2020.

STANOVSKY, G. et al. Supervised open information extraction. In: **Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)**. [S.l.: s.n.], 2018. p. 885–895.

VASWANI, A. et al. Attention is all you need. **arXiv preprint arXiv:1706.03762**, 2017.

WANG, Z. et al. Knowledge graph and text jointly embedding. In: **Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)**. [S.l.: s.n.], 2014. p. 1591–1601.

WU, S.; FAN, K.; ZHANG, Q. Improving distantly supervised relation extraction with neural noise converter and conditional optimal selector. In: **Proceedings of the AAAI Conference on Artificial Intelligence**. [S.l.: s.n.], 2019. v. 33, p. 7273–7280.

XU, P.; BARBOSA, D. Connecting language and knowledge with heterogeneous representations for neural relation extraction. **arXiv preprint arXiv:1903.10126**, 2019.

YE, Z.-X.; LING, Z.-H. Distant supervision relation extraction with intra-bag and inter-bag attentions. **arXiv preprint arXiv:1904.00143**, 2019.

YOO, S.; JEONG, O. Automating the expansion of a knowledge graph. **Expert Systems with Applications**, Elsevier, v. 141, p. 112965, 2020.

ZHANG, Z. et al. Joint model of entity recognition and relation extraction based on artificial neural network. **Journal of Ambient Intelligence and Humanized Computing**, Springer, p. 1–9, 2020.