NER in Archival Finding Aids

Luís Filipe Costa Cunha
University of Minho, Braga, Portugal

José Carlos Ramalho
Department of Informatics, University of Minho, Braga, Portugal

Abstract

At the moment, the vast majority of Portuguese archives with an online presence use a software solution to manage their finding aids: e.g. *Digitarq* or *Archeevo*.

Most of these finding aids are written in natural language without any annotation that would enable a machine to identify named entities, geographical locations or even some dates. That would allow the machine to create smart browsing tools on top of those record contents like entity linking and record linking.

In this work we have created a set of datasets to train Machine Learning algorithms to find those named entities and geographical locations. After training several algorithms we tested them in several datasets and registered their precision and accuracy.

These results enabled us to achieve some conclusions about what kind of precision we can achieve with this approach in this context and what to do with the results: do we have enough precision and accuracy to create toponymic and anthroponomic indexes for archival finding aids? Is this approach suitable in this context? These are some of the questions we intend to answer along this paper.

2012 ACM Subject Classification Computing methodologies → Natural language processing; Computing methodologies → Machine learning; Computing methodologies → Maximum entropy modeling; Computing methodologies → Neural networks; Information systems → Digital libraries and archives

Keywords and phrases Named Entity Recognition, Archival Descriptions, Machine Learning, Deep Learning

Digital Object Identifier 10.4230/OASIcs.SLATE.2021.8

1 Introduction

Throughout the history of Portugal, there was a need to create an archive where information about the kingdom was recorded.

In 1378, during the reign of D. Fernando, the first known Portuguese certificate was issued by *Torre do Tombo* (TT), an institution over 600 years old that is still the largest Portuguese archive, storing a great part of Portuguese historical and administrative records. As time passed, the volume of information contained in national archives has considerably increased, and today there are hundreds of archives spread across the country. Most of these archives have information from the public administration containing records from the 20th century onwards, however, Portugal has three archives with historical information, the *Arquivo Nacional da Torre do Tombo*, the *Arquivo Distrital da cidade de Braga* and the *Arquivo Distrital da cidade de Coimbra* which record various events throughout the history of the country.

The city of Braga was for many years the administrative capital of northern Portugal and of Galicia. In antiquity, most of the records were made by the clergy social class. Even today, the the church's strong influence in the district of Braga is visible, something that influenced the abundance and variety of historical document fonds present in the archive of this district.

At the moment, many of these archival documents are already available to the public in digital format, so it is now intended to interpret their content from a semantic point of view, i.e., to classify and extract different types of Named Entities (NE) present in a

10th Symposium on Languages, Applications and Technologies (SLATE 2021).
Editors: Ricardo Queirós, Mário Pinto, Alberto Simões, Filipe Portela, and Maria João Pereira; Article No. 8; pp. 8:1–8:16
OpenAccess Series in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

given fond. Therefore, this paper proposes the use of entity recognition in natural language, using Machine Learning (ML), a well-known and widely used technique in Natural Language Processing (NLP). In this way, several ML algorithms will be presented, with the intent of generating different results and conclude which algorithm best suits the domain and the problem in question.

2 Related Work

The amount of historical information available in Portuguese archives is increasing, making the exploration of this data complex. Thus, the use of the available computational power is not something new for professional historians. In fact, there are several tools that have been developed over time that assist in the archival data processing.

An example of this is the *HITEX* [17] project, developed by the *Arquivo Distrital de Braga* between 1989 and 1991. This project consisted of semantic model development for the archive historical data, something quite ambitious for that time. Despite this, during its development, it ended up converging to an archival transcription support system, which allowed the transcription of natural text and the annotation by hand of Named Entities enabling the creation of chronological, toponymic and anthroponomic indexes.

Another problem associated with this type of documents was its structure's lack of standardisation. This made it difficult to share information between the archival community both nationally and internationally. To promote interoperability, in Portugal, guidelines for the archival description have been created that describe rules for standardising the archival descriptions [22]. The purpose of these standards is to create a working tool to be used by the Portuguese archivist community in creating descriptions of the documentation and its entity producer, thus promoting organisation, consistency and ensuring that the created descriptions are in accordance with the associated domain's international standards. In addition, the adoption of these guidelines makes it possible to simplify the research or information exchange process, whether at the national or international level.

3 Named Entity Recognition

One of the objectives of NLP is the classification and extraction of certain entities in textual documents. It is easy to understand that entities such as people's names, organisations, places or dates translate into crucial information about certain contexts because this type of data can be used for various purposes, making this practice very popular. Therefore a new NLP subfield rises, Named Entity Recognition (NER).

To be able to recognise entities in texts, two different approaches were taken [10]. Initially, very specific regular expressions were coded to filter various entity types. This mechanism had good results in certain cases where there was an in-depth knowledge of the domain in which this method was intended to be applied. However, this is not always the case, for example, such an approach was considered not to be very dynamic due to the fact that it is necessary to rewrite a large part of the code if one wants to change the domain language. Furthermore, the existence of ambiguity between entities makes them hard to classify, like the names of people, places, etc.

Alternatively, statistical classifiers are used. This method consists of using ML models (this paper will deal with supervised ML models [23]) in order to try to predict whether a certain word sequence represents an entity.

This approach has some advantages over the previous one, such as being able to be used for different languages without having to change a lot of code, the model can be trained with different parameters adjusting to different contexts, an annotated dataset is generated that can be reused for other purposes, etc. In fact, today there are several already trained ML models capable of identifying and classifying various entities, however, the available models are generic, which means that entity prediction for more specific contexts will return results below expectations.

In spite of being much more dynamic than the previous approach, the use of this type of model leads to some work for the experimenter. Thus, it is necessary for the experimenter to write down an annotated training dataset to prepare the model. Despite being tedious work, it has a low complexity level and therefore does not require great specialisation.

4 OpenNLP

One of the tools chosen for this paper was *Apache OpenNLP*, a machine learning-based toolkit implemented in Java, developed by *Apache*. Essentially and as its name implies, its purpose is the processing of natural language through the use of ML algorithms having a wide range of features, such as tokenization, sentence segmentation, part-of-speech tagging, named entity extraction, chunking, parsing, and co-reference resolution [18].

In this paper, the features associated with NER will be addressed, which depend on the tokenization task. At this time, *Apache OpenNLP* provides models for various tasks in several different languages such as English, Spanish, Danish, and some more, however, there is no pre-processed model of NER for the Portuguese language provided by this tool. Available Portuguese annotated datasets like HAREM [6] and SIGARRA [19] were used to train NER models, however, the obtained results were below expectations. In this way, it will be necessary to train one model from scratch.

To understand how *OpenNLP* works, it is necessary to investigate what kind of ML algorithms it uses. In this case, the base algorithm used is Maximum Entropy (MaxEnt).

4.1 Maximum Entropy

We borrow the concept of entropy from physics (thermodynamics) to apply it to various areas of computer science like the *Information Theory* or even classification algorithms, such as MaxEnt where entropy represents the level of uncertainty.

According to [7] in the *Information Theory*, the occurrence of a given event with a low probability of occurring translates into more information than the occurrence of an event with a high probability of occurring. On the other hand, there is *Information Entropy*, which, in *Information Theory* corresponds to the measure of uncertainty, i.e., the average quantity of information required to represent an event drawn from the probability distribution for a random variable. The entropy takes a low value when the probability of certainty for some event is high and it takes a high value when all events are equally likely.

> *"Information entropy is a measure of the lack of structure or detail in the probability distribution describing your knowledge."* [Jaynes, E. T. 1982]

Maximum Entropy Models are statistical models that maximize the entropy of a probabilistic distribution subjected to an N number of constraints. These types of models reveal good results when used to model real-world problems considered hard to model. Usually, they are used on the prediction of high dimensional data, in other words, when there is a much greater number of possible combinations than the amount of available data.

The principle behind this algorithm is that the distribution with the most uncertainty, that is compatible with the context domain, should be chosen. To do so, it is necessary to create several features which represent the information known about the domain. In fact, these features represent restrictions of the model which help the classification of the intended target. After generating the features, it is then necessary to maximize the entropy of all models that satisfy these restrictions. By doing so, we are preventing our model from having features that are not justified by empirical evidence, preserving as much uncertainty as possible. [14]

> *"Ignorance is preferable to error and he is less remote from the truth who believes nothing than he who believes what is wrong."* [Thomas Jefferson (1781)]

4.2 Features

As previously stated, a feature is a way in which known information about the context is passed to the model as constraints, i.e., evidence or hints that make the model correctly classify certain specific cases.

Mathematically speaking, it can be represented as a binary function that for some given $x \in X$, that represents the class of the entities we are trying to predict, and $y \in Y$, that represents the possible contexts that we are observing, it returns the corresponding boolean value.

$$f : X \times Y \longrightarrow \{0, 1\}$$

All features correspond to functions with this signature, however, as already mentioned, a feature represents a constraint, which means that the experimenter must choose the type of information each feature adds to the model, for example:

$$f(a, b) = \begin{cases} 1 & \text{if a = Local and } \textit{checkLocation(b) = true} \\ \\ 0 & \text{otherwise.} \end{cases}$$

$$\textit{checkLocation}(b) = \begin{cases} 1 & \text{if previous word in } b \text{ is "em" and current word starts} \\ & \text{with capital letter.} \\ \\ 0 & \text{otherwise.} \end{cases}$$

In this case, this feature helps the model to classify "`Local`" (place) type entities. In the Portuguese language, when a word beginning with a capital letter is anticipated by the word "em", there is a high probability that this word is a place (e.g., em Braga, em França).

These features are usually context-dependent, i.e., they must be created according to the problem to be modeled. There is often an interdependence between them, making it necessary to iterate over these features so that the decision to be made, in a given iteration, takes into account previous decisions. For example, in the presence of a proper name, it is normal to have a sequence of words that begin with a capital letter. When the first word of the sequence is classified as a person's name, it is very likely that the following words, which start with a capital letter, are also part of that name, so the classifications or decisions made previously are taken into account in current decisions.

As stated in [21], this behaviour about overlapping features is what makes the MaxEnt model really distinguish itself from other models, in the first place, because it is possible to add information already known through features, but also, by letting these features overlap in order to try to predict the best possible results.

4.3 Entropy Maximisation

Following the MaxEnt algorithm, the optimal solution to this classification problem is the most uncertain distribution subject to the defined constraints. The idea behind this is to choose the model that makes the fewer implicit assumptions as possible. Thus, after defining all constraints considered relevant to the context domain, the next step is to maximise the entropy of the model.

To do so, the function of Information Entropy is used.

$$H(X) = -\sum p(a,b)\log p(a,b)$$

The maximum entropy function is a convex function, which means that the value of the weighted average of two points is greater than the value of the function in this set of points. Thus, the sum of the entropy function is also convex. A constraint on this function creates a linear subspace that corresponds to a surface that is also convex and, therefore, has only a global maximum [13].

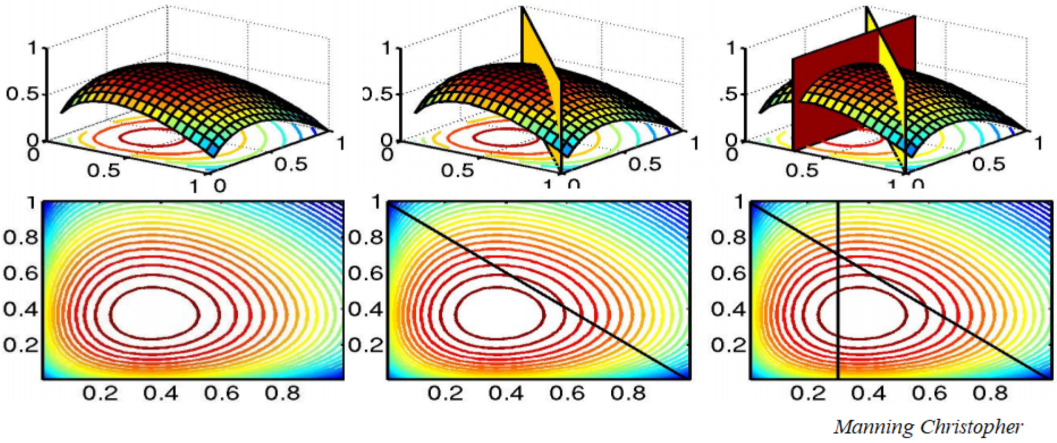


Figure 1 Entropy function subject to restrictions.

As explained in [3], in order to maximise the entropy of the model subject to a limited number of features, it is needed to solve a constrained optimisation problem. In other words, a problem with low complexity can be solved analytically, however, when the number of constraints increases and they overlap with each other, it is not possible to find a general solution analytically. This problem is then solved using Lagrange multipliers by forming a Lagrangian function. An example of this resolution can be found in [15].

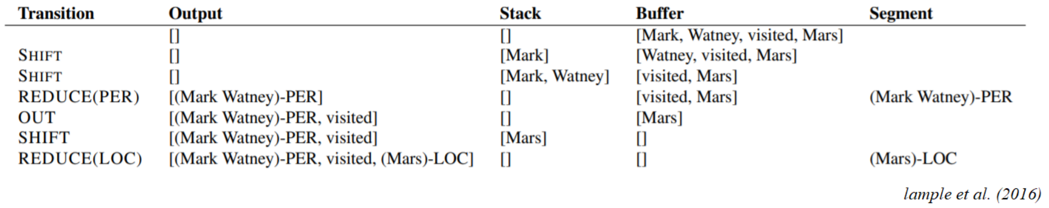
5 spaCy

Another tool that was used in this work is *spaCy*, an open-source library for advanced natural language processing, belonging to the company Explosion, founded by the creators of *spaCy*.

Again, this library offers several features associated with NLP, however, only those relevant to NER will be addressed. Despite having several similarities to *OpenNLP*, *spaCy* presents a very different approach to entity recognition. In addition, the support provided by its creators, documentation and information available about this software is much more accessible. This tool provides several base models of different languages such as Chinese, Danish, Dutch, English, French, Portuguese, etc., which is an advantage over the previous tool. It was implemented in the python programming language and published under the *MIT license.* In paper, the approach taken by *spaCy* regarding entity recognition [24] will be presented.

5.1 Transition Based NER

Most NER frameworks generally use a tagging system, which in practice translates into attaching a tag to each word of interest in the document to further classify it. Instead of using this type of structure, *spaCy* uses a different mechanism to deal with this problem, a transition-based approach.



Transition	Output	Stack	Buffer	Segment
	[]	[]	[Mark, Watney, visited, Mars]	
SHIFT	[]	[Mark]	[Watney, visited, Mars]	
SHIFT	[]	[Mark, Watney]	[visited, Mars]	
REDUCE(PER)	[(Mark Watney)-PER]	[]	[visited, Mars]	(Mark Watney)-PER
OUT	[(Mark Watney)-PER, visited]	[]	[Mars]	
SHIFT	[(Mark Watney)-PER, visited]	[Mars]	[]	
REDUCE(LOC)	[(Mark Watney)-PER, visited, (Mars)-LOC]	[]	[]	(Mars)-LOC

lample et al. (2016)

Figure 2 Example of Transition Based sequence applied on NER.

Analogous to a state machine, this approach is based a set of actions that the model can take in order to make the state machine transit into different states or configurations. The model always takes into account the first word in the buffer and then decides what action it should take. For example, in the Figure 2 we can see that state changes as actions are taken. The challenge of this system is in the prediction of the actions or transitions. In order to address this problem *spaCy* presents a new *Deep Learning* framework [25].

5.2 Deep Learning framework for NLP

In order to predict the actions to be taken in the transition-based model, a statistical model based on Neural Networks is used. The idea starts by finding representations for all words in a given document. After that, it is necessary to contextualise these words in the document, recalculating their representation value. Then the model comes up with a summary vector that represents all the information needed to help predict the target value of the word. From this vector, it is then possible to predict the next valid transition.

To structure this model, the deep learning framework *Embed, Encode, Attend, Predict* divides this whole process into four distinct components, in order to simplify its understanding [25].

5.2.1 Embed

The first task of this approach is *Embed.* This task consists of calculating embeddings using a word identifier, in order to generate vectors for each word in the document.

Figure 3 Embed process.

In fact, the objective of this stage is to generate different representations for words with different semantic meanings through multidimensional vectors. These vectors allow the use of a "hypothesis distribution" so that words that refer to the same entity will have a similar distribution value. This type of mechanism allows the model to be able to associate similar words semantically, even without completely knowing its definition or characteristics, i.e., it does not need to know the word's meaning, but it knows that some words are related in some manner, taking in account the surrounding word vectors. For example, to find out if a particular word refers to a student, words like `human` or `rational being` do not have much impact on entity recognition, however, words such as `study`, `book`, `class` or even `school` are usually related and used near the word `student`. This way the model knows that words like `student`, `pupil`, `finalist` are quite similar in distribution.

This type of mechanism makes the model less limited to the text that was annotated, which translates into a great capacity for learning.

5.2.2 Encode

The encode task aims to transform the vectors previously created in the embedding phase, which are context-independent, and take into account the context in which they are found, thus providing a matrix of context-sensitive vectors.

Figure 4 Encode process.

To make a vector, created from a word, context-dependent, it is necessary to look at the sentence in which this word belongs. For this, in NLP the most common way that several articles address this problem is the use of a Bidirectional Long Short Term memory (BI-LSTM) [11] which takes the whole sentence into account.

However, to determine the context of a vector in the sentence, *spaCy* uses a different method, Convolutional Neural Network (CNN).

This approach only uses four words on either side of a token instead of the whole phrase. Basically, the idea behind this approach is that when using the whole sentence to determine the context of a certain token, the best results are not always obtained, i.e., this practice can make the model show difficulties in knowing if certain context should be associated with the token, making it difficult to discern what matters from what does not. This approach can also provoke the model to over-fit the data, making the model sensitive to things it shouldn't. In this way, *spaCy*'s developer believes that in the vast majority of cases, a small window of words is sufficient to accurately represent the context of a token.

In addition to this, with this type of Neural Network, it is possible to create a decaying effect, to define the level of importance that a given context has on the vector of a word, which is not possible with the previously mentioned method (BI-LSTM).

Finally, it is interesting to note that CNNs have a lower computational cost compared to BI-LSTMs due to the fact that they take advantage of parallelism for each of its layers, managing to use the resources of GPU efficiently [26].

5.2.3 Attend

The third task of this framework is Attend, which consists of taking the matrix built previously and selecting all the necessary information to help the model with the prediction task.

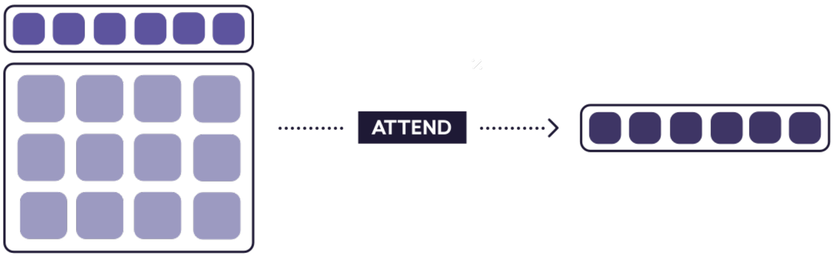


Figure 5 Attend process.

It is at this stage that the model takes into account the defined features. These have a great influence on the way vectors are created, so *spaCy* allows defining them arbitrarily. This type of mechanism ensures dynamism and versatility to the system because depending on the context, it may be necessary to tune the model by modifying the set of features that are being used. At the moment, *spaCy*'s features are defined in the following two steps. Select the first word that is in the buffer, the words that are immediate to the left and right of that word and the last entities previously classified by the model (as the model reads the document from left to right, it is not possible to take into account the entities to the right of the word, because they have not yet been recognised by the model).

After selecting the desired information, a vector that represents problem-specific information is generated and is ready to be used in the prediction phase.

5.2.4 Predict

Finally, there is the last task of this framework, the Prediction.

Figure 6 Predict process.

As its name implies, this step is based on the prediction of the target value. After all the words are turned into vectors (*Embbed*), the vectors are contextualised with the document (*Encode*) and the feature defined are taken into account (*Attend*), the system is ready to make the prediction. This prediction is made using a simple multi-layer perceptron which will return the action probabilities. Then, it is necessary to validate these actions and finally choose the action according to the algorithm's confidence. Finally, this process is iterated through a cycle until the document is finished.

It is important to emphasise that all stages of this framework are pre-computed, that is, they occur outside the cycle, so when the model goes through the document, fewer computations are needed.

6 TensorFlow BI-LSTM-CRF

The *Tensorflow* library, developed by *Google*, presents a vast set of ML features, usually associated with neural networks, which allow to develop and train models in a similar way to the learning method of the human mind. It is an Open Source library, published under *Apache 2.0 license*. By using this library, it is intended to implement a *Deep Learning* model capable of performing NER on archival documents.

Thus, it is necessary to create a system capable of processing the input data, i.e., tokenize the documents and generate word embeddings in order to use them in a specific ML architecture capable of solving sequence tagging problems.

6.1 Recurrent Neural Network

One of the first approaches associated with Deep Learning in NLP was to use Recurrent Neural Networks [8].

In fact, RNNs are famous for obtaining good results on sequential data, which makes them the perfect algorithm for analysing natural text. Despite this, the research community quickly encountered problems associated with this method. First of all, this type of neural network is unidirectional, i.e., it would only take into account the context of the sentence that is before a given token. It is easy to understand that this is a problem when trying to identify a token's entity, because, in order to accurately classify the token, the context of the word's neighbourhood, whether refers to the past or future input, must be taken into account.

In addition, this type of neural network has difficulties in preserving Long Term Dependencies due to the phenomenon of Vanishing Gradient. As the name implies, an RNN has difficulties in preserving contexts observed throughout the sentence, so if there are clues at the beginning of a long sentence, which could help to identify the entity of a token, found at the end of that sentence, they will not be taken into account, which leads to a poor classification.

6.2 Long Short Term Memory

In order to combat RNNs problems, LSTMs are introduced. Basically, LSTMs are RNNs with a memory component added to them in order to create an RNN with Long Term Memory capable of preserving Long Term Dependencies. The introduction of this cell of memory keeps a state that is being updated along the RNN chain. The update of this state is done through gates [16], input, output and forget that regulate the information that must be updated at each timestep, the information that must be passed to the next cell and the information that must be forgotten, respectively. While this state is being updated by the cells of the network, a notion of context is generated for each token, making the RNN capable of taking into account Long Term dependencies.

Using this new method, the Long Term Dependencies problem is solved. However, an LSTM remains unidirectional, so the information that follows a token will not be considered in its classification.

6.3 Bidirectional Long Short Term Memory

A BI-LSTM consists of using two LSTMs, one that will analyse the document in the forward direction, the other backward. The idea behind this architecture is to have an LSTM responsible for capturing the previous context (forward layer), but also, use another LSTM to exploit future context as well (backward layer). In this way, there is information of the entire document associated with each token, contextualising it, something that did not happen with a simple RNN or LSTM. This is only possible due to the ability to preserve Long Term Dependencies resulting from the use of the memory cell otherwise, even if the model analysed the whole sentence it would not be able to maintain all dependencies.

However, although these approaches have better results than a simple RNN, it is important to note that they need more computational resources as it is now necessary to process the memory component in a bidirectional structure.

6.4 BI-LSTM-CRF

For some time, BI-LSTM alone reigned, obtaining state of the art results in several NLP tasks until, in 2015, a new article [9] came out that introduced a new model, BI-LSTM-CRF.

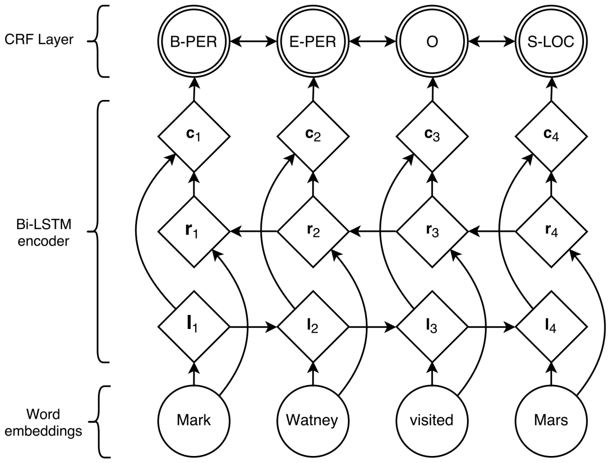


Figure 7 Bidirectional Long Short Term Memory Conditional Random Field.

This architecture consists of using a BI-LSTM and adding a Conditional Random Field (CRF) network to it. This new component receives the consecutive tagging outputs from the BI-LSTM and is responsible for decoding the best tagging sequence, boosting the tagging accuracy [12]. In fact, knowing the relationships between the different labels that one wants to identify in the document, can help the model to classify, with greater accuracy, the best chain of labels. For example, using the BIO format to annotate the phrase "`João Sousa Batista plays Tetris`" we have the following result `[B-Person, I-Person, I-Person, O, O]`. When writing down the name of the person "`João Sousa Batista`" it does not make sense to have the sequence `[B-Person, I-Date, B-Person]`, so this new component comes in a way to help validate the output tag sequence.

The use of BI-LSTM-CRF has already been tested in NLP and has demonstrated state-of-the-art results in various tasks. In this way, this work will use the *TensorFlow* library to implement a solution with this architecture, testing it in archival documents.

7 DataSets

The data used to test the algorithms referred in this article correspond to datasets from two national archives, the Arquivo Distrital de Braga [1] and the Arquivo Regional e Biblioteca Pública da Madeira [2] where the following NE types were extracted: Person, Profession or Title, Place, Date and Organisation.

Firstly, there is a dataset of a fond that shows a pioneering period in computing history, between 1959 and 1998. This fond (PT/UM-ADB/ASS/IFIP), produced by the International Federation for Information Processing (IFIP), contains a section corresponding to the Technical Committee 2, which has a subsection corresponding to Working Group 2.1, that is composed of several series where different archival descriptions are organised, for example, correspondences, meeting Dossiers, news from newspapers, etc.

Secondly, there are two datasets corresponding to a series (PT/UM-ADB/DIO/MAB/006), from the archival fond Mitra Arquiepiscopal de Braga, which contains genre inquiries. The archival descriptions in this series contain witnesses' inquiries to prove applicants' affiliation, reputation, good name or "blood purity". One of the datasets has a very standardised structure, while the other contains a lot of natural text elements.

Thirdly, there is a historical dataset corresponding to the fond (PT/UM-ADB/FAM/ACA) of the Arquivo da Casa do Avelar (ACA) which depicts the family history of Jácome de Vasconcelos, knight and servant of King D. João I. This family settled in Braga around the years 1396 and 1398 with a total of 19 generational lines, up to the present time [1]. This fond is composed of subfonds and subsubfonds that contain records associated with members of this family with a patrimonial, genealogical and personal domain.

Fourth, there is the dataset of the Familia Araújo de Azevedo fond (FAA), also known as Arquivo do Conde da Barca. This archive, produced from 1489 to 1879 by Araújo de Azevedo's family, who settled in Ponte da Barca and Arcos de Valdevez (district of Viana do Castelo) at the end of the 14th century, contains records predominantly associated with foreign policy and diplomacy across borders. This fond is composed of several subfonds composed of archival descriptions with information from members of the FAA family, such as requirements, letters, royal ordinances, etc.

Fifth, there is a dataset that characterizes the streets of Braga in the year 1750. This corpus contains elements that characterize the history, architecture and urbanism of each artery in the city, which help to understand the main lines of its evolution.

Finally, two datasets from the Arquivo Regional e Biblioteca Pública da Madeira were used which correspond to two archival fonds, more precisely to Paróquia do Jardim do Mar and Paróquia do Curral das Freiras, both parishes from Madeira archipelago. These fonds consists of three series each, which represent registrations of weddings, baptisms and deaths. Each series consists of files, that correspond to the year of each record and finally, each file has a set of pieces with archival descriptions.

In total, the annotated corpora contain 164478 tokens that make up 6302 phrases. All the annotated corpus are available to the public in [4]. The distribution of entities is presented in the Table 1.

Table 1 Number of annotated entities per corpus.

Corpus	Person	Place	Date	Professions or Title	Organization	Total
IFIP	1503	325	100	40	318	2286
Familia Araújo de Azevedo	369	450	118	428	94	1459
Arquivo da Casa Avelar	465	239	141	118	91	1054
Inquirições de Genere 1	2002	3713	121	0	0	5836
Inquirições de Genere 2	692	10	54	0	0	756
Jardim do Mar	2393	574	1762	1	2	4732
Curral das Freiras	8729	0	0	0	0	8729
Ruas de Braga	1126	1293	684	391	338	3832
Total	17279	6604	2980	978	843	28684

8 Data Processing

In order to perform entity recognition with *OpenNLP*, *spaCy* and *TensorFlow* in these datasets, it is necessary to train different models so that they learn to find Named Entities in different contexts accurately.

For that, it is necessary to have annotated text that represents each dataset domain. In order to do so, a shuffle of each dataset was performed proceeded with the annotation of a significant fraction of each of them. This shuffle allows the data selection to be impartial, making it a more representative sample of each context domain. During the annotation process, several techniques were used, such as regular expressions, manual annotation and even the use of a statistical model proceeded by correction of the output by the annotator. To facilitate this process, a simple javascript program was created that allows to annotate texts in the browser with a simple keypress.

After the annotation process, the annotated datasets were divided into two parts, 70% of each was used to train the model while the remaining 30% was reserved for validation. It is important to note that the three tools used to implement the NER algorithm use different input data formats. Thus, in this stage, parsers were implemented to convert datasets into three different formats. After that, the tokenization process is initiated. Both *spaCy* and *OpenNLP* have their own Portuguese optimised tokenizer, however, *TensorFlow* does not implement this tool out of the box. In this way, several tokenizers were experimented such as using the Keras tokenizer API, the use of regular expressions and finally *spaCy*'s tokenizer. In this case, *spaCy*'s tokenizer showed better results due to the fact it is optimised for the language in question.

With the datasets processed we feed them into the ML algorithms in order to train the NER models. In this process, individual optimisations are performed for each tool, such as defining the hyperparameters, tunning the models in order to generate the best results.

9 Results

The metrics used to measure NER models' performance are Recall, Precision and F1-score since the accuracy metric does not satisfy the needs of this NLP area [5].

Looking at the Table 2, we can conclude that the created NER models were able to successfully classify most of the intended entities. It appears that in most cases, the BI-LSTM-CRF model generated with *TensorFlow* obtains the best results with an F1-score between 86,32% and 100%, followed by *spaCy* with an F1-score between 70,09% and 100%, and finally *OpenNLP* with an F1-score between 62,67 and 100%. As we can see with these results, the introduction of Deep Learning on NER reveals significant advances in this field.

It is important to note that only one model was created for datasets with high proximity in the context domain. For *OpenNLP*, when using the corpus Genere Inquiries 2 to validate the model trained on the Genere Inquiries 1 dataset, the results obtained were lower (62,67% F1-score) in comparison to the other tools (87,78% and 98,78% F1-score). In this case, it turns out that deep learning has demonstrated a greater capacity for transfer learning.

Finally, analysing the Table 2 we see that the FAA dataset is the one in which the models presented the lowest results. One reason for this is that it contains very long sentences. In fact, as previously seen, a Bi-LSTM-CRF is prepared to deal with Long Term Dependencies which makes it present better results than the other tools (86.32% F1-score).

Table 2 Named Entity Recognition results.

Corpus	Model	Tool	Precision(%)	Recall(%)	F1-Score(%)
IFIP	IFIP	OpenNLP	87,08	82,61	84,79
		spaCy	88,16	89,90	89,02
		TensorFlow	96,12	98,67	97,00
Familia Araújo de Azevedo	Familia Araújo de Azevedo	OpenNLP	72,56	72,30	72,43
		spaCy	74,41	72,82	74,09
		TensorFlow	88,98	87,28	86,32
Arquivo da Casa Avelar	Arquivo da Casa Avelar	OpenNLP	80,15	79,85	80,00
		spaCy	87,82	87,18	87,50
		TensorFlow	89,25	93,25	90,63
Inquirições de Genere 1	Inquirições de Genere 1	OpenNLP	99,93	98,87	99,90
		spaCy	97,35	95,08	96,20
		TensorFlow	100	100	100
Inquirições de Genere 2	Inquirições de Genere 1	OpenNLP	63,17	62,17	62,67
		spaCy	89,66	85,98	87,78
		TensorFlow	98,86	98,95	98,78
Jadim do Mar	Jardim do Mar	OpenNLP	100	99,86	99,93
		spaCy	100	100	100
		TensorFlow	100	100	100
Curral das Freiras	Jardim do Mar	OpenNLP	93,37	99,84	96,50
		spaCy	99,97	99,90	99,93
		TensorFlow	100	100	100

After obtaining the above results, an attempt to create a generalised model with annotations of all datasets was made.

Annotating a fraction of a dataset where this technology is to be applied is not always practical, so it would be interesting to create a generalised model capable of adapting to new contexts of similar nature. On the other hand, it is important that this new model doesn't obtain worse results in the already observed datasets, due to its degree of generalisation.

In this way, the generalised model was trained with 70% of each dataset to be later validated with the remaining 30% of each one. This procedure was repeated for the three tools, obtaining the following results.

Table 3 Generalised NER model validation results.

Corpus	Tool	Precision(%)	Recall(%)	F1-Score(%)
IFIP	OpenNLP	89.43	83.60	86.41
	spaCy	86.99	88.71	87.84
	TensorFlow	92.84	96.85	94.08
Família Araújo Azevedo	OpenNLP	81.94	63.67	71.66
	spaCy	75.19	76.78	75.98
	TensorFlow	78.22	82.47	78.89
Arquivo da Casa Avelar	OpenNLP	88.84	81.68	85.11
	spaCy	87.18	87.18	87.18
	TensorFlow	86.83	92.21	87.99
Inquirições de Genere 1	OpenNLP	99.60	99.53	99.57
	spaCy	98.31	96.74	97.52
	TensorFlow	100	100	100
Inquirições de Genere 2	OpenNLP	74.70	65.61	69,80
	spaCy	79.96	92.21	87.26
	TensorFlow	93.70	98.34	94,82
Jardim do Mar	OpenNLP	99.71	99.71	99.71
	spaCy	99.15	100	99.57
	TensorFlow	100	99.60	99.72
Curral das Freiras	OpenNLP	93.49	99.69	96.49
	spaCy	99.98	99.90	99.94
	TensorFlow	100	100	100

As can be seen, the results obtained by this model are similar to the previous ones, so we can say that the NER performance has not decreased. To measure his performance in a different context, a new corpus with brief notes of the streets of Braga (1750), was annotated and then, after processing it, it was used as validation generating the following results:

Table 4 Generalised NER model validation results on Ruas de Braga corpus.

Corpus	Tool	Precision(%)	Recall(%)	F1-Score(%)
Ruas de Braga	OpenNLP	73.09	61.09	66.55
	spaCy	75.39	62.62	68.42
	TensorFlow	50.50	58.80	53.00

The results obtained are lower than intended. In fact, in addition to the model not having been trained with any part of this dataset, it contains a lot of Organization and Profession type entities. As can be seen in Table 1, the model was trained with few instances of this type, thus the entity recognition may prove challenging.

On the other hand, it appears that the model generated with BI-LSTM-CRF obtained worse results. One of the reasons for this is the fact that this model has a vocabulary reduced to his training data. In fact, both deep learning frameworks presented in this paper represent words through word embeddings, however, *spaCy* uses pre-trained word embeddings from a Portuguese corpus named *Bosque* [20] which makes it have a much larger vocabulary. In this way, *spaCy*'s model can assign semantic meaning to words that were not present in his training which becomes a valuable tool when evaluating corpus from different contexts than the one it was trained on.

10 Conclusion

The archival finding aids used in this paper contain very specific structure and context, which means that available generic NER models may present results that are lower than intended. In addition, there is the language barrier, that is, the amount of Portuguese annotated data, available to train this type of model, is limited.

Despite this, in this paper, it was demonstrated that by training our own models with data that coincides with those we intend to perform NER, it is possible to obtain satisfactory results. In fact, with the advances in the use of Deep Learning in this area of NLP, F1-score values above 86% were achieved in almost all of the used datasets.

Thus, observing the obtained results, it is considered that the use of ML algorithms to perform entity recognition in archival documents, is suitable and with this approach, we can extract information that allows us to create different navigation mechanisms and create relations between information records.

Future work

One way to improve the results presented in this article would be to increase the amount of annotated data. Passing more information to the models' training makes them able to process a greater variety of data making them more generic. On the other hand, it would be interesting to explore new technologies that aim to address this problem, for example, the attention mechanism [27].

Lastly, entity linking could be performed in order to make it possible to browse between different archival documents but related by some entity. It would also be interesting to use the extracted entities to create toponymic and anthroponomic indexes to understand the impact that this tool could have on browsing archival finding aids.

References

1 Archeevo Arquivo Distrital de Braga. Bem-vindo ao arquivo distrital de braga. Accessed in 10-03-2021. URL: `http://pesquisa.adb.uminho.pt/`.

2 Archeevo Arquivo Regional e Biblioteca Pública da Madeira. Accessed in 10-03-2021. URL: `https://arquivo-abm.madeira.gov.pt/`.

3 Adam L. Berger, Vincent J. Della Pietra, and Stephen A. Della Pietra. A Maximum Entropy Approach to Natural Language Processing. *Computational Linguistics*, 1996.

4 Luís Filipe Costa Cunha and José Carlos Ramalho. URL: `http://ner.epl.di.uminho.pt/`.

5 Leon Derczynski. Complementarity, F-score, and NLP evaluation. In *Proceedings of the 10th International Conference on Language Resources and Evaluation, LREC 2016*, 2016.

6 Cláudia Freitas, Cristina Mota, Diana Santos, Hugo Gonçalo Oliveira, and Paula Carvalho. Second HAREM: Advancing the state of the art of named entity recognition in Portuguese. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta, 2010. European Language Resources Association (ELRA). URL: `http://www.lrec-conf.org/proceedings/lrec2010/pdf/412_Paper.pdf`.

7 Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning (Adaptive Computation and Machine Learning series)*. The MIT Press, November 2016. URL: `https://www.xarg.org/ref/a/0262035618/`.

8 Alex Graves, Abdel Rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. In *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 2013. `doi:10.1109/ICASSP.2013.6638947`.

9 Zhiheng Huang, Wei Xu, and Kai Yu. Bidirectional lstm-crf models for sequence tagging, 2015. `arXiv:1508.01991`.

10 Grant S. Ingersoll, Thomas S. Morton, and Andrew L. Farris. *Taming text: how to find, organize, and manipulate it.* Manning, Shelter Island, 2013. OCLC: ocn772977853.

11 Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. Neural architectures for named entity recognition. In *2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL HLT 2016 - Proceedings of the Conference*, 2016. `doi:10.18653/v1/n16-1030`.

12 Xuezhe Ma and Eduard Hovy. End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. In *54th Annual Meeting of the Association for Computational Linguistics, ACL 2016 - Long Papers*, 2016. `doi:10.18653/v1/p16-1101`.

13 Christopher Manning. Maxentmodels and discriminative estimation. URL: `https://web.stanford.edu/class/cs124/lec/Maximum_Entropy_Classifiers.pdf`.

14 OpenNLP Maxent. The maximum entropy framework, 2008. Accessed in 24-09-2020. URL: `http://maxent.sourceforge.net/about.html`.

15 Mike Morais. Neu 560: Statistical modeling and analysis of neural data: Lecture 8: Informationtheory and maximum entropy, 2018. Accessed in 20-10-2020. URL: `http://pillowlab.princeton.edu/teaching/statneuro2018/slides/notes08_infotheory.pdf`.

16 Christopher Olah. Understanding lstm networks, August 2015. Accessed on March 10, 2021. URL: `http://colah.github.io/posts/2015-08-Understanding-LSTMs/`.

17 José Nuno Oliveira. Hitex: Um sistema em desenvolvimento para historiadores e arquivistas. *Forum*, 1992.

18 Apache OpenNLP. Welcome to apache opennlp, 2017. Accessed in 18-10-2020. URL: `https://opennlp.apache.org/`.

19 André Ricardo Oliveira Pires. Named entity extraction from portuguese web text. Master's thesis, Faculdade de Engenharia da Universidade do Porto, 2017.

20 Alexandre Rademaker, Fabricio Chalub, Livy Real, Cláudia Freitas, Eckhard Bick, and Valeria de Paiva. Universal Dependencies for Portuguese. In *Proceedings of the Fourth International Conference on Dependency Linguistics (Depling 2017)*, 2017.

21 Adwait Ratnaparkhi. *Maximum entropy models for natural language ambiguity resolution.* PhD thesis, University of Pennsylva, 1998.

22 Ana Maria Rodrigues, Catarina Guimarães, Francisco Barbedo, Glória Santos, Lucília Runa, and Pedro Penteado. Orientações para a descrição arquivística, May 2011. URL: `https://act.fct.pt/wp-content/uploads/2014/05/ODA-3%C2%AA-vers%C3%A3o.pdf`.

23 Satoshi Sekine and Elisabete Ranchhod. *Named Entities: Recognition, classification and use.* John Benjamins Publishing Company, July 2009.

24 spaCy. spacy 101: Everything you need to know · spacy usage documentation. Accessed in 07-01-2021. URL: `https://spacy.io/usage/spacy-101`.

25 spaCy. Model architecture, 2017. Accessed in 14-01-2021. URL: `https://spacy.io/models`.

26 Emma Strubell, Patrick Verga, David Belanger, and Andrew McCallum. Fast and accurate entity recognition with iterated dilated convolutions. *CoRR*, 2017. `doi:10.18653/v1/d17-1283`.

27 Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017-December, 2017.