Resolving Named Entities and Relations in Text
for Applications in Literary Studies

Joana Rocha
joana.rocha@tecnico.ulisboa.pt
Instituto Superior Técnico
Avenida Rovisco Pais
1049-001 Lisboa, Portugal

Abstract. Lately, there has been an increase of texts available in digital libraries. There is also an increased interest in capturing semantic relations expressed between entities from a large amount of texts, hoping to conduct more complex semantic tasks, such as question answering. However, traditional information extraction techniques have a hard time following this sudden trend, as these techniques rely heavily on manually annotated resources for training statistical models. This research work had one main objective: to adapt and evaluate two relation extraction systems that follow a new Information Extraction paradigm, usually referred to as Open-Domain Information Extraction (OIE), in order to extract relations from Portuguese literary texts. This new information extraction technique is able to scale to a massive amount of texts, without requiring as much human involvement. The two systems in focus are named ReVerb and OLLIE. Many tasks were addressed in order to obtain the results presented in this document, starting with the development of NLP models to process Portuguese texts, followed by the incorporation of these models into the OIE tools, and further changes in implementation details, resulting in two adapted systems that are able to process Portuguese texts. This document, therefore, formalizes the approaches in the relation extraction problems, involving the new OIE paradigm on literary texts, presenting an extensive evaluation with four different literary books.

Keywords: Text Mining, Open-Domain Information Extraction, Relation Extraction, Natural Language Processing.

1 Introduction

For the past few years, there has been an enormous increase in the amount of texts available in digital libraries. Avid readers and Humanities’ scholars, researching about literary texts and trying to answer theoretical questions about its content, consequently have their productivity damaged as they cannot accompany this sudden trend. Recent developments have also led to the use of various techniques from the area of Information Extraction within distant reading approaches, for instance facilitating the literary analysis and the automatic construction of knowledge bases. This document addresses the subject of Information Extraction, namely by focusing on a novel extraction paradigm called Open Information Extraction. Traditional information extraction techniques rely on extensive human involvement in the form of hand-written rules or hand-tagged training examples. Open Information Extraction systems, instead, are able to scale to massive and heterogeneous texts, without requiring as much human involvement as traditional information extraction techniques. Literary texts evidence these challenges as they can be very large and of different genres.

My MSc thesis concerns with the general problem of Information Extraction on works of fiction books, envisioning applications in literary studies and the extraction of semantic knowledge between fictional entities on large documents. The language of the literary texts in focus will be Portuguese, unlike the majority of previous work developed in the field which has focused mainly on English texts. My research work tried to prove, through controlled experiments, the following hypothesis: Open Information Extraction techniques can be effective in extracting information, namely valid relations between characters and locations, in Portuguese literature texts. The remainder of this document is organized as follows: Section 2 presents the basic concepts within the field of Natural Language Processing. Section 3 presents previous publications, reporting techniques used to extract information in literary texts. It also presents several IE systems which use the new Open Information Extraction paradigm to extract relation phrases and the arguments in Web texts. Section 3 concludes with some related work focusing on the use of Information Extraction techniques on Portuguese texts, as the other presented previous studies have been applied only to English texts. Section 4 describes the development processes which originated the necessary products in order to proceed to an evaluation. Section 6 concludes this document by reviewing the work done throughout this MSc thesis, and discusses some possible ideas for future work.
Natural Language Processing (NLP) is an interdisciplinary field concerning with tasks that involve the use of a human language [16]. Most systems relying on the output of NLP tasks start by receiving plain text and then process it by making it go through a pipeline of operations. These operations then return valuable information that is later accessed in order to infer more information from it.

Operations such as sentence splitting and tokenization, take the input document represented as a plain text and later divide into sentences and words. The resulting sentences and tokens will be the basic units needed for other NLP operations, such as POS tagging, NER and dependency parsing. These tasks can also be referred as sequence tagging classifiers as they receive an input, usually a string of tokenized words representing a sentence, and output the sequence of labels that best suit each string unit based on a statistical model. The following are commonly used NLP tasks described in a briefly manner.

- **Part-of-speech (POS) tagging** is the process of labeling each word of a given sentence with its corresponding lexical category [16]. POS taggers label each word according a limited set of lexical categories.
- **Named-Entity Recognition (NER)** is one of the most important NLP tasks within the context of Information Extraction. It has the goal of finding named entities in a text, and classifying them into semantic categories such as Person, Organization or Location.
- **Chunking** is a partial parsing method, used when one is not in need of a complete parse tree. Non-overlapping segments, called chunks, correspond to the major part-of-speech tags found in each grammar rule. Noun-phrase chunking, in particular, consists of finding the noun phrases — phrases with a main noun and some modifiers — in a text, treating others indifferently.
- **Dependency Parsing** presents one way of describing the structure of sentences. In this structure, individual words are connected to each other. Each link, called a dependency, holds two lexical nodes (e.g., words) and is drawn from a fixed inventory of labels, which represent grammatical functions. One verb is usually the root of the sentence, not depending on any other word. A word only depends on another if either is a complement or a modifier of the latter.
- **Relation Extraction** is a task that has the goal of capturing semantic relations in text. A relation is defined as an association between or among things. The association is usually expressed as an action, through a verb, between two entities, e.g. Sarah lives in Lisbon. Other associations between entities can be found where these are not mediated through a verb, i.e. implicit relations, and other associations involve more than two arguments, i.e. a $n$-ary relation. Relations are usually described and extracted as binary relations $t = (c_1, r, c_2)$ in which there are two arguments and a relation phrase defined between them.

To implement the aforementioned NLP tasks, supervised approaches following machine learning algorithms are typically used. Sentences are usually modeled as sequences of tokens and given as input to sequence classifiers. A sequence classifier is a model that receives, as input, a sequence of single units, i.e. tokens, and outputs the sequence of labels that best classifies each unit. POS tags, named-entity classes and chunks are some examples of labels used in sequence classifiers.

Hidden Markov Models (HMM), Maximum Entropy Markov Models (MEMM), Conditional Random Fields (CRF) are some examples of commonly used statistical models that typically perform the sequence classification tasks. A corpus is required in order to build these models, which is composed of sentences with each word correctly labeled. The annotated corpus will hereafter supply the model with gold answers, from which are extracted additional data to help the model detecting the outputting labels. This information usually takes the form of a feature vector which is composed of linguistic attributes that describe each token and its surrounding context. Features are then extracted based on the training corpus, containing annotated sentences.

A training corpus can also manifest an encoding style. This is a particularity often found in training corpora for tasks such as NER and Chunking, as these contain multi-word segments and encoding styles help define the boundaries of these segments. The two most frequent encoding styles are the BIO and the SBIEO scheme.

### 3 Related Work

Most IE approaches that focus on literary texts try to extract interactions between characters and places when gathering information from them. These approaches usually express the gathered information in the form of a social network, where different nodes (i.e. entities) are connected whenever a textual semantic relationship is found between them. This general approach became popular with Franco Moretti and his definition of distant reading, which he defines as a method that tackles literary problems, namely focusing on plot analysis, by scientific means.

Other studies followed Moretti’s plot analysis. For instance Elson and McKeown (2010) [10] developed an approach that automates the study of a large sample of literary texts, more specifically 60 nineteenth century novels from various categories. His goal was to verify some theories about the social world of nineteenth century novels.
ficiton, focusing mainly on character’s interactions.
Even though literary studies center on characters and their direct interaction, there is semantic knowledge outside quoted speech. For example, in the Genesis book, there are 330 distinct person names but only 53 of those are involved in dialogue interactions. In this situation, the previous method would only be able to capture a small part of the character’s relationships. Lee et al. [17] tries to extract social networks from texts that lack dialogue interactions.
Quoted text is a particularity that some documents present, such as literary texts or news articles. Most work on quoted speech has focused on the news domain, but quotes can come in various syntactic forms. Approaches in literary texts try to assign a speaker to each quote found in text [11][15].
Some previous approaches have instead focused on extracting specific relations from literary texts. One particular approach [18] tried to used the ReVerb system to extract family relations on novels. Despite many relations obtained, none of them captured family relationships between characters. Therefore, a new approach was created by combining word level techniques and utterance (i.e. quote) attribution approaches.
Open-Domain Information Extraction is an Information Extraction technique, better suited when the types of relations to be extracted are not known a priori. OIE systems follow two major branches of techniques when extracting relations and try to achieve a good overall performance since these two techniques pose a trade-off between accuracy and processing time.
Shallow features refer to properties of a word, such as its POS tag, while dependency parsing captures the relations between words and the structure of the sentence. TextRunner [2] pioneered the alternative IE paradigm of open domain extraction, by being the first scalable and domain-independent OIE system. TextRunner uses shallow features in order to extract relations as it uses POS tagging and NP-chunking to perform the extraction process.
Subsequent work was done involving a new model of TextRunner, where instead of a Naive Bayes classifier it uses a liner-chain CRF. This improved system, called O-CRF [3], proved to be better by achieving 88.3% in precision and 45.2% in recall. O-CRF’s training process is self-supervised. It applies some relation-independent heuristics to the Penn Treebank and obtains a set of labeled examples in the form of relational tuples. Similar to the Naive Bayes TextRunner system, it uses these sets of examples to extract features and then train the CRF model. Given an input, O-CRF also does a single pass over the data, performing the POS tagging and NP-chunking.

WOE [23] is another OIE system, although different from TextRunner as it automatically transfers knowledge from the Web, namely from Wikipedia and DBpedia pages, in order to be able to extract relations without limitations. WOE has another particularity: it disposes of two types of extractors. WOE\textsuperscript{parse} depends on shallow features, to extract bootstrapping tuples for the final extractor, while WOE\textsuperscript{parse} relies on dependency features. Articles from Wikipedia are used for the bootstrapping process, where infoboxes’ attributes are matched to sentences in the article. These resulting tuples then train each extractor, which will then be able to perform the extraction process. While WOE\textsuperscript{parse} provides more accurate results, it requires more time to process a single sentence compared to the other systems.
Previous aforementioned OIE systems cannot fully capture some relation phrases, showing incoherent and uninformative extractions. ReVerb [12] is an OIE system that eliminates these two issues by introducing syntactic and lexical constraints. The syntactic constraint imposes that the relation phrases should have a defined sequence of POS tags although this pattern can sometimes create overly specific relation phrases. The lexical constraint eliminates overly specific extractions, maintaining a dictionary of relation phrases seen with at least 20 distinct arguments. ReVerb is a shallow feature system as it uses POS tagging and NP-chunking in order to perform the extractions. ReVerb’s performance was fairly higher compared to previous mentioned systems, achieving an area under precision-recall curve 30% higher than WOE\textsuperscript{parse}.
ReVerb and WOE share two important weaknesses:
1. They only extract relations mediated by verbs;
2. Both ignore context, thus often extracting tuples that are not asserted as factual, as they perform a local analysis of the sentence.
OLLIE [21], Open Language Learning for Information Extraction, is an OIE system that overcomes the limitations of previous systems by expanding the syntactic scope of the relation phrases to cover a much larger number of relation expressions, i.e. capturing relation phrases mediated by non-verbs. It also expands the OIE representation to allow additional context information, such as attribution and clausal modifiers. Since ReVerb’s verb-based expression is capable of covering a broad range of relations, OLLIE uses high confidence seed tuples generated from ReVerb. OLLIE’s second step is to learn the patterns that encode various ways of expressing relations, called open pattern templates. Overall, OLLIE has a bigger area under precision-yield curve, about 2.7 times larger than ReVerb and 1.9 times larger than WOE\textsuperscript{parse}.
In contrast to the previous techniques, other studies [6][7] have been made in which semantic roles were used
for the task of OIE. SRL-IE [6] is a Semantic Role Labeling (SRL) based extractor in order to extract relation tuples. SRL-IE uses UIUC-SRL [19] as a base system for performing SRL, converting its final output into a more similar format to that from OIE systems.

Only few OIE systems have specifically dealt with Portuguese texts. One of the approaches found in the literature was a multilingual OIE system [14] that overcomes some of the challenges found while doing unsupervised extractions, such as extracting other non-verbal mediated relations and events with more than two arguments. Another approach [4] proposed to find the most similar relations from a set of previously annotated ones. The procedure consists on a classifier based on a nearest neighbor classification (kNN) where each training example has a weight associated, measuring the similarity compared to the instance being classified.

4 Resolving Named Entities and Relations in Literary Texts

In order to evaluate the hypothesis, two current state-of-the-art OIE systems were modified. These systems are the ones with currently the best trade-off in terms of accuracy and performance, namely ReVerb and OLLIE. Stanford’s NLP frameworks were used in order to be able to train and use the each required NLP models. A POS tagger, NP-chunking and dependency parsing model were developed and later incorporated into the two OIE systems. A NER model was also developed, using the same framework, in order to extract the named-entities from literary texts.

Several datasets were used in the construction of these models, and preprocessing steps were done in order to be able to minimize the performance errors when merging these. However, these details are better described in Section 5. This Section details how the two OIE systems in focus, namely ReVerb and OLLIE, were adapted to the Portuguese language, along with the development and incorporation of task-specific NLP models in these systems.

4.1 POS Tagging

Stanford’s POS Tagger framework\(^1\) was used in order to train and evaluate the POS tagging model. In brief, its structure is similar to a bidirectional dependency network. Bidirectional approaches use the next and previous tag. This works by making an initial estimation using observations of the local environment, then proceeding in using the referred bidirectional dependency network. Finally, a variation of the Viterbi algorithm is used to identify the maximizing sequence \([22]\).

The training datasets for POS tagging came from the merge of four datasets, namely CINTIL, Floresta Sintática, Tycho Brahe and a text made available by the Universal Dependencies project contributors. These many datasets were required, not only due to the different nature of each dataset but also due to the vast universe of content words that can appear in each literary text.

Features in the training process involved not only basic tagger features such as the suffix and prefix of words, but also non-annotated resources. These additional resources were composed by all four datasets in their textual form in order to build word clusters. Clusters were built using a open-source implementation\(^2\) of the Brown clustering procedure and tuned according to the results obtained on different NLP tasks [9].

4.2 Entity Recognition

The supplied corpus used to train the NER model was the CINTIL corpus. This corpus considers four types of entities/categories. Three of these categories are the standard Person, Location and Organization followed by a fourth Miscellaneous category. The corpus uses the SBIEO encoding in order to establish the boundaries of each entity, as its generally the encoding that give best results on NER models [20].

Stanford’s NER framework\(^3\) was the chosen tool to train and evaluate the models. It uses a liner chain Conditional Random Field sequence model, coupled with feature extractors. Besides basic features made available by Stanford’s NER framework, the use of resources based on non-annotated text was also considered. These additional features are as follows:

- Word Clustering: Clusters were built using open-source implementation of Brown clustering procedure. Texts used to train the word clusters consisted in the merge of the CINTIL corpus (non-tagged) with a document containing news published over the course of 10 years, from the Público newspaper;
- Stems: The use of the stem form of each word present in corpus, calculated by using SNOWBALL Stemmer\(^4\), a tool that uses the Porter algorithm in order to obtain the stem of a word;

\(^1\) [http://nlp.stanford.edu/software/tagger.shtml](http://nlp.stanford.edu/software/tagger.shtml)
\(^2\) [https://github.com/percyliang/brown-cluster](https://github.com/percyliang/brown-cluster)
\(^3\) [http://nlp.stanford.edu/software/CRF-NER.shtml](http://nlp.stanford.edu/software/CRF-NER.shtml)
\(^4\) [http://snowball.tartarus.org/algorithms/portuguese/stemmer.html](http://snowball.tartarus.org/algorithms/portuguese/stemmer.html)
Gazetteers and name lists: Two name lists were used, each of them containing common female and male names taken from Wikipedia lists. Gazetteers were also used, containing names that belong to either people, locations and organizations. These were built by extracting both real and fictional entities from Freebase;

Upper-cased: A new source of information, calculated with all the content words from the CINTIL corpus.

This new data consists on a value indicating the amount of times a word was seen with the first letter upper-cased, 1 if verified in all corpus and 0 otherwise.

4.3 NP-Chunking

Building the chunker model involved two frameworks which resulted in two final models. Stanford’s Parser framework\(^5\) developed the main chunker model and another model was mistakenly built using OpenNLP’s framework\(^6\). The latter derived from the assumed inability of Stanford’s Parser to return NP-chunks. However, both models were used for NP-chunking. OpenNLP’s chunker model is able to process sentences much faster, and it was used for heavy-computation tasks only, such as ReVerb’s dictionary building task.

Three datasets possessed manually annotated syntactic tags: CINTIL, Floresta Sintáctica and Tycho Brahe. The final dataset used for OpenNLP’s training process was composed by all of these three datasets merged together. Stanford’s Parser final training dataset was only the Tycho dataset, as this one revealed a better consistency of sentence structure and was able to detect more NP-chunks compared to the other datasets.

In order to build both models, the constituency tags were transformed into broader tags and a normalization process took place.

4.4 Dependency Parsing

Once again, the selected approach to develop this task model was provided by Stanford’s Parser framework. This framework provided two ways to construct models that would be able to parse Portuguese text and return its dependencies. The chosen approach relies on the use of CONLL formatted training files, in which the model is trained by giving it direct dependency examples. This approach\(^5\) uses a transition-based dependency parser along with a neural network classifier to make the parsing decisions. The transition-based parser uses a greedy algorithm, aiming to predict a final sequence of transitions (i.e. arcs) that lead to a dependency tree. It employs the standard-arc system that consists in a configuration \(c = (s, b, A)\), where \(s\) is the stack, \(b\) the buffer, and \(A\) the set of dependency arcs. The parser shifts the words from the buffer to the stack, while the neural network predicts the correct transition based on a set of words from the stack and buffer, and its corresponding POS tags and arc labels. The transition-based dependency parser ends when the buffer \(b\) is empty and the stack \(s\) contains only the single node \([ROOT]\), return \(A\) as the final set of dependencies.

Training the dependency parser model involved two datasets in the CONLL format. Both these datasets were made available by the Universal Dependencies project contributors. Training process required word embeddings in order to improve the performance of the parser. Word embeddings were created with the word2vec\(^7\) tool using all the available datasets plus the set of news from Público and the Wikipedia Portuguese articles. This tool created and mapped \(d\)-dimensional vectors to each word of the vocabulary.

4.5 Adapting ReVerb

ReVerb tries to avoid the major errors coming from the previous systems by relying its extraction process on two main rules. These extraction rules are the core of ReVerb, as it aims to fully capture relation phrases and avoid incoherent and uninformative extractions. While adapting ReVerb, the first step consisted in altering the syntactic rule. This rule focuses on a part-of-speech-based regular expression that defines the sequence of POS tags that relation phrases should comply with. The original rule was therefore changed in order to use the Universal Tagset, with the help of the available treebank mappings\(^8\).

The second step was then to incorporate the POS tagger model and the NP-chunking suited models. Since all tools were developed using the same programming language, this was not a complicated task and the models were incorporated easily.

The first rule aims to capture the relation phrases fully, while the second rule, the lexical rule, tries to avoid the extraction of overly specific relation phrases by using a dictionary. The dictionary is composed of various relation phrases in its stem form and the respective number of different arguments these were seen with. Before returning an extraction, ReVerb checks the dictionary to see if the current relation phrase was seen with at least

---

\(^5\) http://nlp.stanford.edu/software/lex-parser.shtml

\(^6\) https://opennlp.apache.org/documentation/1.5.2-incubating/manual/opennlp.html

\(^7\) https://code.google.com/archive/p/word2vec/

\(^8\) https://github.com/slavpetrov/universal-pos-tags
with 20 (default parameter) different second arguments. The dictionary had to, therefore, be rebuilt as part of
the adaptation as it would reflect on the lexical rule of the adapted system. In order to accomplish this task,
all available datasets were gathered, including a set of news text from Público and all Portuguese articles from
Wikipedia. ReVerb's stemmer was also replaced for open-source Portuguese stemmer\(^9\).
Finally the confidence function was re-trained using around 200 sentences taken from Portuguese Wikipedia
articles, expressing relations between the subject entity and the attributes from the infoboxes. ReVerb processed
this set of sentences and the resulting extractions were manually annotated as either correct or incorrect.

### 4.6 Adapting OLLIE

OLLIE relies heavily on dependencies and therefore, the first step in adapting this tool was to incorporate
the dependency parsing and POS tagging models. Due to incompatibilities between versions of programming
languages and the runtime environment, the tagger and parser had to be incorporated in a Java wrapper. This
wrapper uses the tagger and parser to extract the dependencies of the input sentences and feeds OLLIE with
this information in the right format. The wrapper then receives the extractions and returns these to the user.
The second step consisted in altering specific code and other resources, such as:

- Modification of the dependency labels and tags stated in code in order to reflect and match the most recent
  set (the Universal Dependencies and the Universal Part-of-speech Tagset). This was accomplished with the
  help of the available tag mappings\(^{10}\) and older dependency version/Stanford Dependencies mappings to
  more recent ones [8];
- Alterations in language-specific code that affected mostly the open pattern templates' building. OLLIE
  generates the open pattern templates by firstly building patterns and then the templates. The patterns
  refer to a simple representation of dependency paths between the arguments and the main content word
  of the relation phrase (usually a verb), while the templates is a simpler representation of the relation
  phrase only, encoding the words with a broader labels. As an example, the following are the templates and
  corresponding patterns of a simple extraction, in both before and after adaptation:

  **Before:**
  ```
  be {rel} {prep} <nsubjpass< {rel:postag=VBN} >{prep:regex=prep,(\.)}⟩ > {arg2}
  ```

  **After:**
  ```
  ser {rel} {nmod} <nsubjpass< {rel:postag=VERB} >{nmod:regex=nmod,(\.)}⟩ > {arg2}
  ```

  The adapted OLLIE encodes the auxiliary Portuguese verbs (i.e. ser, estar, haver, ter) by transforming the
  relation phrase words into their stem form and comparing them with a list of auxiliary verbs. It also encodes
  the prepositions that typically appear merged with the dependency nmod. This happens due to the Stanford
  CCProcessed dependencies format which simplifies the dependency tree by merging the non-content words
  of the parse tree, mostly auxiliary words such as prepositions;
- Switched the current English Stemmer for a open-source Stemmer\(^{11}\) that creates stems from Portuguese
  words.

OLLIE is able to extract relation tuples by following a model that contains patterns and templates, expressing
not only the relation phrase composition but as well as the association with the dependency path between the
arguments and the relation. These are called open pattern templates, and are saved in a file along with its
confidence level.

After altering language-specific code, it was then possible to create the model that contains these open pattern
templates. The seed tuples came from high-confidence results of the adapted ReVerb (confidence over 90%) on
all the datasets available. These resulting tuples were then filtered, preserving only the tuples that contained
entities in both their arguments.

In order to obtain a more diverse set of tuples, all content words from the high-confidence set of tuples were
extracted and matched against all datasets. Potential erroneous tuples were eliminated by filtering them if their
dependency path was above 4. Open pattern templates were generated from the merge and processing of these
two resulting sets.

Finally, the confidence function was re-trained using around 200 sentences taken from Portuguese Wikipedia
articles. Each resulting extraction was then manually annotated as either correct or incorrect. The annotated
extractions were then given to the confidence function where new weights were calculated from a set of features.

### 5 Results

This section presents the final results and analysis related to the testing of NLP models and the two adapted
OIE systems, as well as the alterations done in the training datasets.

\(^9\) [https://code.google.com/archive/p/ptstemmer/](https://code.google.com/archive/p/ptstemmer/)

\(^{10}\) [https://github.com/slavpetrov/universal-pos-tags](https://github.com/slavpetrov/universal-pos-tags)

\(^{11}\) [https://code.google.com/archive/p/ptstemmer/](https://code.google.com/archive/p/ptstemmer/)
5.1 Datasets and Methodology

In order to merge and test the NLP models developed, preprocessing steps were needed in which annotations details had to converge. This section enumerates the major preprocessing decisions done during the development of the NLP models.

All datasets had their tags and labels changed. The POS tagset used in the development of the models belongs to the Universal Part-of-speech Tagset\(^\text{12}\) while the label set used for the dependencies belongs to the Universal Dependencies\(^\text{13}\). Both universal sets consist of harmonized sets of tags/labels that exist across multiple languages, therefore facilitating the development of multilingual tools.

Further alterations were done in the annotation structure, where all contractions were joined together (e.g. de and a resulted in da), as well as all the verbs and their clitics (e.g. *encontrar*- and *-se* were joined in order to produce *encontrar-se*).

All training datasets used and their statistics are presented in Table 1.

5.2 Experimental Results

The NLP models were evaluated using a 5-fold cross validation technique where results were compared using different sets of features.

The POS tagger results are shown in Table 2. Due to the tagger being hard coded, mostly local features were allowed to be used during the training process. Even though the Stanford POS tagger allows the bidirectional approach, i.e. taking into account the tags that follow the current word being tagged, the unidirectional approach showed better results when applied on the final merged dataset. The word clusters feature slightly improves the results, possibly due to being a non-local feature that only affects rare words. Having less rare words compared to common ones, the impact on the results will be minimal.

The NER model was built and tested using CINTIL corpus as the training dataset. Unlike the tagger framework, the NER framework disposes a lot more feature tweaking to train the model, providing the setting of local and non-local features. Table 3 presents several tests done while varying the set of features. The results show that the stems feature really improved the results of the NER. On the other hand, the use of word clusters did not. Facing the statements that NER results improve when increasing the number of clusters \([9]\), we can perhaps conclude that the clusters were still in insufficient number for this task or contains errors. The rest of the features show a slight improvement on the results, however the true effect might be hidden by the errors of the word clusters.

Stanford’s constituency models were evaluated individually on the classification of NP-chunk tags, using 5-fold cross-validation. Through pattern matching, the NP-chunks belonging to the gold-answers and the obtained tree were extracted. NP-chunks were then compared by checking their ranges. If both ranges matched, the NP-chunk obtained would be considered correct. This process was custom developed in order to evaluate these datasets. Although CINTIL possesses higher performance (Table 4), it also possesses smaller training sentences compared to the other datasets. Tycho was the chosen dataset due to its structure consistency and higher number of NP-chunks found after a closely performed evaluation that used a sample of sentences.

OpenNLP’s chunking model was evaluated using 5-fold cross-validation. The training dataset was composed of three datasets in total. Results showed 91.11%, 92.03% and 91.57% in precision, recall and \(F_1\)-measure respectively.

Two datasets were used in order to train the dependency parsing model. Both were selected after evaluating them on a sample of test sentences that belonged to OLLIE tests. The idea was to check if the dependencies did not diverged much from the expected ones, since OLLIE heavily relies on dependency parsing. Results showed a higher number of correct dependencies from the model containing the two datasets, therefore the decision to maintain both in the final model.

Tests done with high-dimensional vectors manifested bad results, reaching 12% in UAS. Best results were achieved by using 50-dimensional word vectors, reaching 79.98% UAS and 75.91% LAS. Results were obtained by testing the two datasets using a 5-fold cross-validation technique (Table 5).

ReVerb and OLLIE were the two systems adapted in the course of this thesis, resulting in these two system being able to process and extract Portuguese relation tuples, following the recent Open Information Extraction paradigm. Both systems were tested on a set of sentences from four different books on public domain:

- *Os Maias*, written by Eça de Queirós;
- *Orgulho e Preconceito*, written by Jane Austen, translated by Lúcio Cardoso;
- *Amor de Perdição*, written by Camilo Castelo Branco;
- *Alice no País das Maravilhas*, written by Lewis Carrol, translated by Isabel De Lorenzo.

\(^{12}\) https://github.com/slavpetrov/universal-pos-tags

\(^{13}\) http://universaldependencies.org
Most related work that focused in Relation Extraction in literary texts, were more concerned in capturing the plot and the character’s interactions. Therefore, tests will focus on the extractions that capture a relation between two entities.

A set of 40 sentences was selected for the tests. From these 40 sentences, half are in agreement with Reverb’s extraction constraints and the other half were randomly selected sentences. The following step consisted in manually extracting and annotating relation tuples of the form \((entity_1, relation, entity_2)\) from this set of sentences. The following types of relations were found during this process:

- Relations following ReVerb’s limitations, where the relation phrase appears explicitly between two entities and these are contiguous;
  Example: \(\text{Quando minha sobrinha Georgiana foi para Ramsgate no Verão passado, fiz questão de que dois criados homens a acompanhassem.}\)
- Relations where the phrase structures are non-contiguous, often having verbs in later positions of the sentence referring to long distant arguments;
  Example: \(\text{Depois de jantar Carlos percorreu o Fígaro, folheou um volume de Byron, bateu carambolas solitárias no bilhar, assobiou malagueñas no terraço - e terminou por sair, sem destino, para os lados do Aterro.}\)
- Implicit relations. These mostly arrive from special phrase structures that always appear between commas, and often specify details that are interesting to capture. These type of relations are not mediated through a verb, unlike the ones mentioned previously.
  Example: \(\text{A este tempo, Manuel Botelho, cadete em Bragança, destacado no Porto, licenciou-se para estudar na Universidade as matemáticas.}\)

Results from the first test are shown in Table 6, using ReVerb in its default form (syntactic constraint and lexical constraint enabled) on the set of sentences. Results show the evaluation of obtained extractions which their level of confidence was higher than 50%. As it can be seen, the first test did not reveal good results. After assessing the failed extractions, it was clear that most were incomplete as they failed to capture the second entity. This is due to the dictionary filtering out long and too specific relations which would lead to the capture of the wrong NP-chunk. Reviewing ReVerb’s dictionary, where the most frequent relation phrases appear at the top, it was clear that it contained mostly short-sized relation phrases.

Since the manually extracted relations, from each book, revealed longer relation phrases, which were not covered in the dictionary, a second test was performed having the lexical constraint disabled. Table 6 shows better results corresponding to the second test. As expected, ReVerb was able to capture more relations tuples (given that these are in contiguous phrase structures) by disabling the dictionary. The frequent punctuation also plays a big part, as it helps defining the boundaries of the obtained extractions. Other extractions keep failing due to NP-chunking errors and relation phrases not matching the syntactic constraint.

NLP task errors were expected. However, errors related to the syntactic constraint deserve more attention. After matching all the manually extracted relations against the syntactic constraint, some failed due to mostly the
Table 3: NER results obtained from testing CINTIL corpus with a 5-fold cross-validation technique.

<table>
<thead>
<tr>
<th>Entity Spans</th>
<th>PER</th>
<th>LOC</th>
<th>ALL</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>P</td>
<td>R</td>
<td>F₁</td>
</tr>
<tr>
<td>Basic</td>
<td>93.64</td>
<td>87.22</td>
<td>90.31</td>
</tr>
<tr>
<td>+ Stems</td>
<td>99.12</td>
<td>98.14</td>
<td>98.63</td>
</tr>
<tr>
<td>+ Word Clusters</td>
<td>97.09</td>
<td>93.26</td>
<td>95.14</td>
</tr>
<tr>
<td>+ Gazetteers</td>
<td>97.29</td>
<td>93.75</td>
<td>95.49</td>
</tr>
<tr>
<td>+ Upper-cased</td>
<td>97.09</td>
<td>94.10</td>
<td>95.57</td>
</tr>
</tbody>
</table>

Table 4: Constituent parsing results of each datasets, using 5-fold cross-validation.

<table>
<thead>
<tr>
<th>Datasets</th>
<th>P</th>
<th>R</th>
<th>F₁</th>
</tr>
</thead>
<tbody>
<tr>
<td>CINTIL</td>
<td>72.29</td>
<td>75.08</td>
<td>72.95</td>
</tr>
<tr>
<td>Floresta Sintáctica</td>
<td>68.22</td>
<td>67.11</td>
<td>67.31</td>
</tr>
<tr>
<td>Tycho Brahe</td>
<td>66.05</td>
<td>66.48</td>
<td>65.48</td>
</tr>
</tbody>
</table>

prepositions in the relations phrases. There were numerous cases were these were at fault, mainly due to the following reasons:

- As it was mentioned in Section 5.1, contractions were merged as a step of the normalization process of the datasets. Contractions are mostly composed of prepositions and a determinant, resulting in a determinant tag being assigned to the resulting merged word. Having these merged, and tagged as a determinant, damaged ReVerb’s performance as a number of long relation phrases did not had the required preposition to end them. Determinants were found instead of the wanted prepositions;
- Other relation tuples would contain more than one preposition, which would make the relation phrase break off earlier than what was supposed to. For example, the sentence *Manuel Botelho mudou de regimento para Lisboa* contains two prepositions, resulting in the following incorrect extraction *(Manuel Botelho; mudou de; regimento)*.

Although changing the syntactic constraint is possible and easily accomplished, the amount of sentences, in which two prepositions appeared in a relation phrase, was significantly lower compared to the other issue found. One can also argue if the change would be beneficial, since it could bring longer relation phrase compositions, to the already long ones found during the manual annotation. Having two prepositions found in a relation phrase can also present evidences that ReVerb cannot fully capture these kind of relations that employ a less direct writing style, for example:

*Elizabeth foi levada até a carruagem por Mr. Collins.*

 Obtained: *(Elizabeth; foi levada até; a carruagem)*

 Expected: *(Elizabeth; foi levada até a carruagem por; Mr. Collins)*

Table 5: Dependency parser results varying datasets, using 5-fold cross-validation.

<table>
<thead>
<tr>
<th>Datasets</th>
<th>UAS</th>
<th>LAS</th>
<th>Out-of-vocabulary Words (% of testing corpus)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Floresta Sintáctica</td>
<td>84.21</td>
<td>80.23</td>
<td>10.75</td>
</tr>
<tr>
<td>Universal Dependencies</td>
<td>82.44</td>
<td>78.78</td>
<td>8.01</td>
</tr>
<tr>
<td>Floresta + UD</td>
<td>79.98</td>
<td>75.91</td>
<td>7.95</td>
</tr>
</tbody>
</table>
Nonetheless, the major problem consisted in wrong normalization decisions.

Moving ahead to OLLIE tests, the testing procedure was similar to ReVerb’s procedure. The same set of sentences is given to OLLIE and extractions are returned. Table 6 shows the results obtained.

As we can see, results are not uplifting. One aspect that contributed to this bad performance was certainly the Dependency Parser’s model lower performance. OLLIE relies heavily on dependency parsing to perform the extractions. It builds open pattern templates which represent the dependency links between the arguments and the relation phrase by means of a simpler dependency labeled structure. Extraction is then performed by transforming the input sentences into their dependencies, followed by the matching of the patterns against the obtained dependencies. Errors can propagate, since one malformed pattern can induce several wrong extractions. Also, badly obtained dependencies match into the wrong, or none, pattern.

After further inspection of the obtained results, it was noticeable that a high number of extractions failed due to details related to the datasets annotations. Often, wrong extractions such as 

\{(rel) \{arg\} \{nsubj\} (\{rel:postag=VERB\} \{obj\} \{arg\})\} 

were marked as subjects of verbs located in later positions of the sentence. This particularity persisted from the original datasets as the normalization process did not affect any dependency labels related to the subject.

Besides the high number of wrong extractions, it was also noticeable that the results had low recall. As it was mentioned, OLLIE returns extractions according to a set of open pattern templates. These patterns have an associated confidence which define the probability of seeing that type of extraction in each sentence. According to new bootstrapping set developed, extractions would belong to at most two extraction patterns. The rest of the patterns had a confidence lower than 9%.

\{(rel) \{nsubj\} (\{rel:postag=VERB\} \{obj\} \{arg\})\} \{nmod:regex=nmod\} \{arg\} 0.6170

Unfortunately, no more tests were performed as OLLIE had core issues related to the accuracy of the dependency parser and a particularity that both datasets showed. Often words such as ‘que’ and ‘onde’ were marked as subjects of verbs located in later positions of the sentence. This particularity persisted from the original datasets as the normalization process did not affect any dependency labels related to the subject.

By comparing the results of both tools, it cannot be concluded that these tools are yet suitable to extract relation tuples from literary texts. ReVerb’s tests showed that this tool was only able to capture relations when in suitable conditions, leaving the rest uncaught. However, OLLIE was able to capture more types of relations, but errors related mostly to parsing errors lead to a worse performance than ReVerb.

### 6 Conclusions

This document presented the details related to the development of this MSc thesis. In the end, several NLP models were developed, capable of processing Portuguese texts, using datasets which are in agreement to the
Universal Part-of-speech Tagset and Dependencies Guidelines. The adaptation of two relation extraction systems under the new OIE paradigm brought an interesting contribution, as there is not much work concerning the use of Open Information Extraction on Portuguese texts, specially on Portuguese literary texts. Across this document, concepts and related works in the field of relation extraction were revisited, followed by the work accomplished in order to obtain the tools needed for evaluation. Finally, the results were presented, from the extensive evaluation related to the development of this thesis. Unfortunately, it was not possible to conclude that the adapted systems were fit for relation extraction on literary texts. ReVerb reached a better performance although it could only extract relation tuples in specific conditions. OLLIE has a much higher potential of accomplishing the task at hands, but errors on the dependency parsing task worsened the results compared to ReVerb.

6.1 Future Work

Despite the results obtained, there are more and new interesting approaches to try out for future work in the area of Relation Extraction in Portuguese texts. One interesting approach would the the use of Semantic Role Labeling to extract relation tuples [6][7]. Although Propbank and FrameNet resources have had a significant increase in amount, Portuguese resources exist in a lower amount compared to resources dedicated to the English language. Nonetheless, this approach could be very useful for extracting relation tuples in literary texts as it is a very powerful tool.

Another interesting approach would be to use translation in order to take full advantage of existing tools. Since the English language has a wider variety of tagged resources and linguistic tools to process English texts, it would interesting to use translation tools in order to obtain the desired output by using English available linguistic tools on non-English texts. Converting non-English texts into English texts, followed by the information extraction procedure and, finally, projecting the output back to the source language, has been tested before with promising results [13]. This approach allows a more complete experimental validation in the field, as it opens doors to the various currently available relation extraction systems which are only focused for one particular language.

Finally, other resources are worth being mentioned as potentially helpful resources in similar developments: (1) Polyglot-NER, not only contains named-entity annotations but also provides a system that can build named-entity annotators for 40 major languages using Wikipedia and Freebase [1]; (2) Colonia Corpus of Historical Portuguese, contains several POS tagged historical texts from the 16th to the 20th century; (3) Corpus Informatizado de Textos Portugueses Medievais, a corpus containing several texts ranging from the 19th to the 21st century; and finally, (4) Mac-Morpho, a corpus containing POS tagged Brazilian texts.

References


14 https://sites.google.com/site/rmyeid/projects/polyglot-ner
15 http://corporavm.uni-koeln.de/colonia/inventory.html
16 http://cipm.fcsh.unl.pt/gencontent.jsp?id=4
17 http://nilc.icmc.usp.br/macmorpho/


