Co-Reference Resolution in Portuguese and Spanish Texts

Nádia Fernandes
Instituto Superior Técnico,
University of Lisbon, Portugal
nadia.sofia@tecnico.ulisboa.pt

Bruno Martins
Instituto Superior Técnico,
University of Lisbon, Portugal
bruno.g.martins@tecnico.ulisboa.pt

Henrique Cardoso
Faculdade de Engenharia,
University of Porto, Portugal
hlc@fe.up.pt

Abstract

Co-reference resolution is a task focused on identifying the expressions in a text referring to the same entity. It has attracted a great deal of attention due to its importance in language understanding and as a subtask for other Natural Language Processing problems. The current state-of-the-art approaches are based on the supervised training of deep neural networks, which presents a challenge for less-resourced languages, such as Portuguese. In this paper we propose a state-of-the-art neural co-reference resolution model for Portuguese and Spanish texts. The developed model explores a cross-lingual learning approach, aligning Portuguese and Spanish word embeddings in a single vector space and training simultaneously with data from both languages, tackling the problem of Portuguese being a less-resourced language. Our model builds on a previous neural co-reference resolution system, developed and tuned for English data, which we adapt to the cross-lingual scenario.

Keywords

Co-Reference Resolution, Cross-Lingual Learning, Deep Learning, Natural Language Processing

1 Introduction

Natural Language Processing (NLP) is a scientific field focused on giving computing machines the ability to process, interpret and generate natural language (i.e. human language). Hence, NLP involves several tasks, such as the co-reference resolution task. Co-reference resolution consists in identifying the expressions in a text (e.g. pronouns and nouns) that refer to the same entity. Those referring expressions are called mentions and the entity to which they refer to it is called the referent. A group of mentions with the same referent is called a co-reference chain or a cluster. The goal of a co-reference resolution system is to output all the co-reference chains of a given text.

The problem of co-reference resolution has been studied for many years. However, recently, researchers started testing cutting-edge deep learning techniques to solve it, re-gaining the interest in the subject. Co-reference resolution already has several existent solutions trained and tested over English texts, but few experiments with neural models have been done for Portuguese, as a less-resourced language. For this work we propose a model for co-reference resolution on Portuguese and Spanish texts. We explore a cross-lingual learning approach, aligning Portuguese and Spanish word embeddings in a single vector space and using data from both languages to train. Our co-reference resolution system is based on a previous Neural Network (NN) approach developed for English data, aligned with the state-of-the-art - NeuralCoref[1]. The code developed during this work is available online[2].

This paper presents the following structure: Section 2 introduces related work on co-reference resolution, focusing on systems developed for Portuguese and Spanish texts. Section 3 describes the proposed NN model, its hyperparameters and training strategy, and the cross-lingual word embeddings used. Section 4 details the experimental evaluation, specifying the datasets and their pre-processing, the evaluation metrics, and the obtained results. Finally, Section 5 summarizes the conclusions and presents suggestions for future work.

2 Related Work

Over the years, several works have been developed for co-reference resolution (Sukthanker et al., 2018). Initial approaches worked on rule based resolution, using hand-crafted rules based on syntactic and semantic features of the text. Over the years, co-reference resolution shifted to machine learning approaches (e.g. using decision trees) and recently to deep learning models, relying on NNs. The latter models present the best results on the task and correspond to the state-of-the-art. Stylianou and Vlahavas (2019) presented a review on several neural models, such as the ones developed by Clark and Manning (2016) and Lee et al. (2017), achieving high results for English data.

Regarding co-reference resolution for Portuguese data, Fonseca et al. (2017a) presented CORP (Co-Reference Resolution for Portuguese), a rule based approach using lexical, syntactic and semantic knowledge. The model has a multi-step architecture, applying a rule in each step and grouping two mentions if the restrictions are satisfied. They start by performing mention detection using CoGrOO parser (Silva, 2013), a grammar checker which also provides syntactic annotations. Then, a set of 13 rules is applied: 11 lexical and 2 semantic. The lexical rules cover exact and partial matches, appositive constructions, abbreviations, nominative predicates and relative pronouns. The semantic rules cover hyponymy and synonymy relations, obtained using Onto.PT (Oliveira, 2012).

[1] https://github.com/huggingface/neuralcoref

[2] https://github.com/NadiaSofia/Co-reference-Resolution-for-PT-and-ES.git

CORP links a mention to its antecedents if some rule is verified. Based on these co-reference pairs, clusters are formed. However, in some cases a mention can be linked to antecedents from different clusters (i.e. referring to different entities), so it is necessary to decide which cluster the mention belongs to. In this situation, CORP would erroneously output a single cluster with all mentions. To tackle that clustering problem, Fonseca et al. (2018) proposed a clustering method which takes into account discourse structure, using the CORP model as baseline. They assume that any mention is new in the discourse if it does not have a link to one or more antecedents. Thus, the clustering algorithm works as follows: if the mention does not have any co-reference relation, a new cluster is created; if the mention only has a co-reference relation with one cluster, it is linked to that cluster; if the mention has co-reference relations with more than one cluster, a clustering criteria is applied to decide to which one it is linked.

As a clustering criteria, Fonseca et al. (2018) presented the five options: Closest Cluster, Cluster Weight, Mention Weight, Mention + Cluster Weight and F1-Score Weight. Given a mention m to link: the cluster weight is obtained summing +1 for each CORP rule satisfied by the co-referent mentions to m from that cluster; the mention weight is obtained summing +1 for each co-referent mention to m; the F1-Score weight of a cluster is obtained summing the weight of each CORP rule satisfied by the co-referent mentions to m from that cluster, where the weight of each rule corresponds to the CoNLL F1-Score obtained by applying it individually.

Both systems presented were tested on the Portuguese dataset Corref-PT (Fonseca et al., 2017b).

Regarding co-reference resolution for Spanish data, some works were developed for the SemEval-2010 Task 1 (Recasens et al., 2010). They were developed on a gold scenario regarding mention detection, using the gold mention boundaries from the dataset. The dataset used was AnCora-CO-ES (Recasens and Martí, 2010), which was already divided in training and test sets.

Kobdani and Schütze (2010) proposed SUCRE, an approach based on a relational database model and a regular feature definition language. Its architecture is divided in two parts: pre-processing and co-reference resolution. In pre-processing, the text corpus is modeled to a relational database model, which involves extracting atomic word features, detecting markables (i.e. mentions) and extracting atomic markable features. Atomic features are attributes - examples of atomic word features are the position of the word in the corpus, the gender and number, and the Part-of-Speech (POS) tag; examples of atomic markable features are the number of words in the markable, the syntactic role and the semantic class. Having the relational database model, co-reference resolution can be performed. Considering this, SUCRE has five functional components:

1. Relational Database Model of Text Corpus: requires at least the Word, Markable and Link tables.
2. Link Generator: for training, it generates a positive instance for each co-referent markable pair and negative instances for a markable and all its not co-referent antecedent markables.
3. Link Feature Extractor: link features are defined over a pair of markables. A regular feature definition language with some keywords and functions is used to select different word combinations of the two markables.
4. Learning: trains a Decision Tree classifier on the train data.
5. Decoding: applicable to test data. Creates the clusters using best-first clustering - searches for the best antecedent (i.e. the one with the highest probability) predicted as co-referent.

Sapena et al. (2010) developed RelaxCor, a co-reference resolution system based on constraint satisfaction. It represents the problem as an undirected graph connecting any pair of candidate co-referent mentions. Given a pair of mentions, a set of constraints restricting their compatibility is used to compute the weight of the edge connecting them. Each constraint has a weight associated, reflecting its confidence. The edge weight is the sum of the weights of the constraints that apply to that mention pair. The weights can be positive or negative, indicating whether the mentions are co-referent or not, respectively.

The constraints are learned automatically by evaluating a set of features over each pair of mentions in the training data. The features used are lexical, morphological, syntactic, semantic, and about distance and position. The learned constraints are conjunctions of feature-value pairs, forming a positive example if the considered pair of mentions is co-referent, and a negative one otherwise. The weight associated with each constraint is the fraction of co-referent examples where the constraint applies minus a balance value.

RelaxCor uses relaxation labeling over the set of constraints for the resolution process. Relaxation labeling is an iterative algorithm that performs function optimization based on local information. It maintains a vector with a probability distribution for each mention, where each value corresponds to the probability of the mention belonging to a specific partition (i.e. entity) given all the possible partitions. During the resolution process, these probability vectors are updated taking into account the edge weights and the probability vectors of the adjacent mentions. The larger the edge weight, the greater the influence of the adjacent probability vector. The algorithm updates the probability vectors in each step until convergence. The final partitioning is directly obtained by assigning each mention to the partition with the highest probability.

Attardi et al. (2010) proposed TANL-1, a co-reference resolution system using a binary classifier and a greedy clustering technique. A Maximum Entropy classifier is trained to determine whether two mentions refer to the same entity or not. Regarding the training instances, a positive instance is created by pairing each mention with each of its co-referent antecedents, and a negative instance is created by pairing each mention with each of its preceding non co-referent mentions. For each pair of mentions, the classifier is trained using lexical, distance, syntax, type (namely Named Entity and Pronoun types), gender and number features. According to the output of the classifier, the mentions are

		MUC			B^3			CEAF_e			CoNLL
		P	R	F1	P	R	F1	P	R	F1	Avg F1
ES	TANL-1	56.5	16.6	25.7	93.4	65.2	76.8	64.7	66.9	65.8	56.1
	RelaxCor	**73.8**	14.8	24.7	**97.5**	65.3	78.2	66.6	66.6	66.6	56.5
	SUCRE	58.3	52.7	**55.3**	79.0	75.8	77.4	69.8	69.8	69.8	67.5
	Arch-BiLSTM	42.7	**65.7**	51.6	72.2	**86.6**	**78.8**	**87.5**	**72.3**	**79.2**	**69.9**
PT	CORP	44.2	52.2	47.9	35.8	45.8	40.2	46.1	43.9	44.9	44.3
	CORP+Clustering	**54.9**	50.2	**52.5**	**51.8**	43.6	47.3	46.2	**52.8**	**49.3**	**49.7**
	Arch2	46.8	**59.7**	**52.5**	47.0	**62.6**	**53.7**	**55.1**	34.5	42.4	49.5
Direct Transfer (ES-PT)	Arch2	**56.9**	**60.9**	**58.7**	**58.6**	**39.7**	**45.8**	**33.0**	**28.0**	**29.7**	**44.8**

Table 1: Evaluation results for co-reference resolution on AnCora-CO-ES and Corref-PT datasets.

clustered using best-first clustering.

All the previously presented models were developed on a monolingual scenario and using rule based or machine learning approaches. Not many co-reference resolution neural models have been developed for Portuguese or Spanish, nor models exploring cross-lingual learning for these languages. However, recently, Cruz et al. (2018) proposed a state-of-the-art system for co-reference resolution on Spanish and Portuguese data, exploring a cross-lingual setting: direct transfer learning from Spanish to Portuguese.

Their work was focused on the classification phase of co-reference resolution, so the model was supplied with gold mention boundaries, not dealing with mention detection. The linking algorithm used was *closest antecedent*, which links each mention to its closest positively identified antecedent, if there is one.

They subdivided the proposed neural models in two steps: (i) extracting representative features for mentions, which is performed using Convolutional Neural Networks (CNNs), Long Short-Term Memory networks (LSTMs) or dense layers; (ii) assessing co-reference affinity, which is performed using dense layers. Given this, they tested five different model variations:

1. Arch1: composed by an embedding layer whose vectors were obtained from a pre-parsed FastText file, containing the most common words at training; word embeddings are summed getting a mention embedding; the embeddings from both mentions are stacked and passed through a standard 1D convolutional layer; the obtained representation is concatenated with scalar distance features and passed through two fully-connected layers: a standard one and a final sigmoid-activated one, which is the output layer.
2. Arch2: the embedding layer is created by tokenizing all the texts from the input dataset, loading the entire embeddings model and using FastText ability to predict embeddings for the out-of-vocabulary words; remaining layers are still the same as Arch1.
3. Arch2-dense: the embedding layer is the same as Arch2; resulting embeddings are also summed and then concatenated with the distance features; the obtained vector is passed through two hidden layers and the final output layer.
4. Arch-deep-CNN: the embedding layer is the same as Arch2; to obtain the mention representation, instead of summing the word embeddings, their vectors are passed through two 1D convolutional layers; the obtained vectors are max-pooled along the first axis, and then passed through two hidden layers and the final output layer.
5. Arch-BiLSTM: the embedding layer is the same as Arch2; then, the word embeddings are fed into a Bidirectional LSTM layer; the last state of the LSTM is extracted and passed through two hidden layers and the final output layer.

All hidden and convolutional layers are activated by a ReLU function. For the word embeddings, they used pre-trained FastText multilingual word vectors (Grave et al., 2018), whose vector spaces were aligned after training, meaning the Portuguese and Spanish versions of a word have close vector representations in their respective embedding spaces.

To obtain the training instances, Cruz et al. (2018) created pair-wise combinations of mentions by pairing each mention with all its candidate antecedents. An instance is created for every pair of mention and antecedent, adding a third element specifying their co-reference relation: $(mention, antecedent, P)$ if positively co-referent, or $(mention, antecedent, N)$ if not. Since this generates a highly unbalanced dataset, with more non co-referent instances, they used a random undersampling of that class. The undersampling percentage which was able to maximize the model performance was 70%.

The two datasets used were AnCora-CO-ES for Spanish, and Corref-PT for Portuguese. The AnCora-CO-ES corpus was already split, so the results were reported on the test set, the development set was used for validation and the training set was used to train the models. On the other hand, the Corref-PT corpus was not split, so they randomly selected 60% for training, 20% for development and 20% for testing.

Table 1 summarizes the results on the MUC, B^3 and CEAF_e metrics, and the CoNLL score for the systems presented in this section. For CORP+Clustering (Fonseca et al., 2018), we report the results using the Cluster Weight criteria, which presented the best results. Similarly, for the work of Cruz et al. (2018) we only report results for the architecture with the best performance. Additionally, exploring a cross-lingual scenario, Cruz et al. (2018) experimented direct transfer of model weights from Spanish to Portuguese by training the model on the Spanish data and testing it on the Portuguese test set. The results of the architecture with the best performance for direct transfer learning are also reported in Table 1.

3 Co-Reference Resolution for Portuguese and Spanish Texts

This section presents the developed model, which is an adaptation of a previous one: NeuralCoref. Figure 1 shows the overview of our system.

NeuralCoref is an extension for SpaCy, implementing a state-of-the-art neural co-reference resolution system. It uses NNs to resolve co-reference clusters and includes simple contextual information in mention representations. Its co-reference resolution algorithm is divided in three steps: (i) extracting a series of mentions (pronouns, noun phrases and named entities); (ii) computing a set of features for each mention and pair of mentions; (iii) finding the most likely antecedent for each mention (if there is one) based on the set of features.

The first step is performed by a rule based mention detection function, using SpaCy Tagger, Parser and Named Entity Recognition (NER) annotations to identify the potential mentions. The mention detection rules defined in NeuralCoref were specific for English texts, so for our model we modified this function. We will explain it in detail in Subsection 3.1.

Once all the potential mentions are identified, the model reaches the second step: extracting a set of features for each mention and each pair of mentions. This is performed using word embeddings and some additional integer and boolean features. To include some simple contextual information about the mentions in the features, the model takes embeddings for several words inside and around each mention and averages them, generating span vectors.

The single mention features are:

- The type of the mention (e.g. noun, etc.);
- If the mention is nested;
- The type of the document (e.g. news, etc.);
- The location of the mention;
- The size of the mention;
- Indices for the word embeddings of the following mention elements: root, first word, last word, previous word, next word, second previous word, second next word, root head;
- Span vectors for the following elements: mention, five words before the mention, five words after the mention, sentence, document.

For a mention pair, the features are:

- If the mentions have the same speaker;
- If the mention speaker's name is in the antecedent;
- If the antecedent speaker's name is in the mention;
- If there is an exact string match between them;
- If there is a relaxed string match between them (i.e. nouns/proper nouns match);
- If there is a match between the mentions' roots;
- The distance between them;
- The sentence distance between them;
- If the mentions overlap.

Finally, these features are concatenated and fed into two NNs, entering the third step. The model has a common embedding layer that transforms the words embedding indices (one of the mention features explained above) in word vectors, before feeding the NNs. The first NN computes a score for each pair of a mention and a possible antecedent, taking as input the single mention features for each mention, along with their pair features. The second NN computes a score for a mention having no antecedent, taking as input its single mention features. All these scores are compared and the highest determines if the mention has an antecedent and if so, which one. The training goes through three successive phases: All-Pairs, Top-Pairs and Ranking. The model moves on to the next stage when the established number of epochs is over or if the evaluation metric on the development set stops increasing for three epochs. When changing to the next stage, the best model from the previous stage is loaded. The first phase, All-Pairs, uses a cross-entropy loss on the full set of mention pairs. The second phase, Top-Pairs, also uses a cross-entropy loss but only on the top scoring antecedents (true and false) of a mention. The last phase, Ranking, uses a max-margin loss with slack-rescaled costs.

NeuralCoref was developed to process English data (CoNLL-2012 corpus), so for our model, in addition to changing the rules for mention detection, we adapted it to read the Portuguese and Spanish corpora and to work in a cross-lingual scenario, using it as base model.

3.1 Mention Detection

For the mention detection task, we relied on SpaCy models for each language, providing POS Tagger, Parser and NER annotations. This information is used on a set of rules to identify candidate mentions. For Portuguese we used the *pt_core_news_sm* model and for Spanish we used the *es_core_news_sm* model. We started by parsing the corpus documents with SpaCy. From the SpaCy parsed doc, we were able to extract spans (i.e. phrases), which we analyzed one by one to obtain the candidate mentions - pronouns, noun phrases and named entities.

For each span, we went through each token applying the following set of rules:

1. Verify if the coarse-grained POS tag corresponds to a noun (NOUN), a proper noun (PROPN) or a pronoun (PRON), or if the syntactic dependency label corresponds to a nominal subject, an indirect object or an object. If this does not verify, move on to the next token.
2. If the token is a personal or relative pronoun, add it as a candidate mention.
    i. If the pronoun is part of a conjunction, obtain the span that goes from its leftmost to its rightmost syntactic descendants and add it as a candidate mention.
    ii. Move on to the next token.
3. Obtain the leftmost and rightmost token's syntactic descendants. Take the span that goes from the left one to the right one and clean it, by verifying if it does not start or end in punctuation, conjunctions, interjections or prepositions. If the start (or end) is not valid, move it to the next (or previous, respectively) position, until reaching a valid span or an empty one. If a valid span is obtained, add it as a candidate mention. Otherwise, move on to the next token.
4. Given the previously obtained span, verify if there

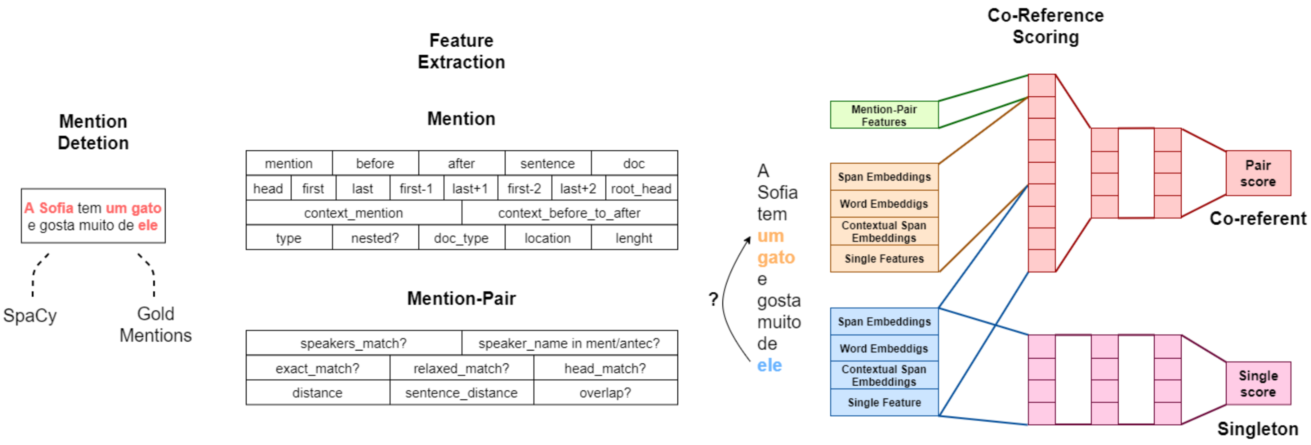

Figure 1: Overview of the co-reference resolution system proposed.

is punctuation in it. If so, separate a segment containing the token being processed, whose limits correspond to the closest punctuation (on each side), not including them (e.g. previous span: *o presidente, Marcelo Rebelo de Sousa*, initial token: *presidente*, separated segment: *o presidente*). Clean the separated span (as explained above) and, if it is valid, add it as a candidate mention. Otherwise, move on to the next token.

5. Given the previously obtained span, verify if there are conjunctions or prepositions in it. If so, separate a segment containing the initial token, whose limits correspond to those terms (the closest ones), not including them (e.g. previous span: *o gato e o cão*, initial token: *cão*, separated segment: *o cão*). Clean the separated span (as explained above) and, if it is valid, add it as a candidate mention. Otherwise, move on to the next token.
6. Finally, if the span obtained from the previous steps starts and/or ends with a verb, remove the verb (e.g. previous span: *a equipa comandada*, new span: *a equipa*). Clean the obtained span and, if it is valid, add it as a candidate mention.

We started by implementing rules 1, 2, 3 and 5 as a base. Comparing the obtained candidate mentions to the gold ones, we observed that a considerable number of candidate mentions contained punctuation in them and some smaller gold mentions within those were not recognized as candidate mentions. We covered those cases by adding rule 4 to the previous ones. We also noticed that some examples like the one described in rule 6 were happening, in which some candidate noun phrases included a verb classifying the noun, but the gold mentions did not, so we added that rule, increasing the number of correct mentions detected on both languages.

In addition to these rules, SpaCy is able to recognize Named Entities (NEs). For Portuguese and Spanish, the accepted entities can represent a named person or family (PER), a name of politically or geographically defined location (LOC), a named corporate, governmental, or other organizational entity (ORG), and miscellaneous entities - events, nationalities, etc. (MISC). In both languages, we added those NEs as candidate mentions. Analyzing some examples from each language dataset separately, we concluded that some additional specific rules could be added.

For Spanish, we noticed that the used dataset contained underscores as mentions to missing subjects (e.g. *Lo peor que* [_] *podemos hacer a un menor de edad*). Most of the times, those underscores were identified as proper nouns or syntactically classified as nominal subjects or objects, as they were supposed to. However, going through the rules, sometimes the candidate mentions recognized did not include the underscores individually, only spans with them surrounded by other words. To overcome this, we added a rule to verify if the token being processed was an underscore, and if so we added it as a candidate mention.

For Portuguese, we observed that our function was considering parts of a proper noun as candidate mentions, along with the complete proper noun (e.g. candidate mentions: [[*Maria*] [*Joana*]], gold mention: [*Maria Joana*]). Since the dataset does not consider these cases as mentions, we added a rule verifying if a candidate mention corresponding to a proper noun was contained in another one, also corresponding to a proper noun. If so, the contained mention was removed from the candidate mentions, reducing the number of wrong candidates generated. We also noticed that in examples like *a atleta Maria* (a nominal modifier followed by a proper noun) our rules detected [*a atleta* [*Maria*]] as candidate mentions, instead of [*a atleta*] [*Maria*] as in most of the gold annotations. To fix this, we added a rule verifying if a candidate mention corresponding to a proper noun was contained in another candidate mention. If so, we took the part differentiating them (i.e. *a atleta*) and cleaned it. If the obtained span was valid, we added it as a candidate mention. Despite this generating several more candidate mentions, it also adds more correct candidates, maintaining the ratio between them. This complete set of rules can generate duplicates, so by the end we cleaned the candidate mentions to eliminate them.

3.2 Cross-Lingual Word Embeddings

Cross-lingual approaches to word embeddings attempt to unify language representations, trying to get similar representations according to the words' meaning regardless of their language. For our proposed model, we focused on developing a cross-lingual system for Spanish and Portuguese, so we used the MUSE library[3] to obtain the cross-lingual word embeddings.

The library already provides embeddings for Portuguese and Spanish aligned on a single vector space; however, they only make available a text file with the embeddings of the words they used in the alignment. Since it is not possible to obtain the embeddings for out-of-vocabulary words and not all words in our training data are present there, we redid the alignment. As there is more Spanish data available, it was used as the source language and Portuguese was used as the target language for the alignment. We used the Supervised method available in the MUSE library, which takes a bilingual dictionary and learns a mapping from the source to the target space using (iterative) Procrustes alignment. The library already had available a bilingual dictionary for Portuguese and Spanish (obtained using the Unsupervised method), which we used to align our embeddings. We chose the Supervised method over the Unsupervised one since all the necessary resources were available and the former is faster.

In addition to the bilingual dictionary, the Supervised method also takes as input text files with embeddings for each language. Those files must contain a very considerable amount of samples (i.e. word embeddings), so we can obtain a good alignment. To obtain those input files, we started by getting pre-trained FastText word vectors for each language[4]. The word vectors were available in two formats: a text file and a binary model, which can be used to obtain vectors for out-of-vocabulary words. We gathered all the FastText word embeddings in the text file, and added the ones corresponding to the out-of-vocabulary words (present in the training data and not in the file), loading the binary model and using FastText ability to predict their embeddings. We did this for both Portuguese and Spanish, using the obtained files as input for the MUSE model. The model outputs a text file with the Portuguese MUSE embeddings and another one with the Spanish MUSE embeddings, aligned in the same vector space. The embeddings in each output file represent the words given in the corresponding input file.

Since we used a lot of words for the alignment, in the end we filtered each output file obtained, including just the words in the respective training data, using those versions in our model.

3.3 PCA Projection

Following the work of Mu et al. (2017), we decided to do a post-processing for the word embeddings, in which the objective was to obtain more discriminative representations. The post-processing consists in eliminating from the word vectors: the common mean vector and a few top dominating directions, obtained through Principal Component Analysis (PCA). The idea behind it is that all word representations share a same common mean vector and have the same dominating directions. Such vector and directions strongly influence the word vectors in the same way, so by eliminating them the representations capture more discriminative information.

Given the word representations $\{r(w), w \in V\}$, V being the vocabulary and w a word, the post-processing algorithm goes as follows:

1. Compute the mean of the word representations: $\mu = \frac{\sum_{w \in V} r(w)}{|V|}$
2. Update the word representations by subtracting the mean from them: $\tilde{r}(w) = r(w) - \mu$
3. Using the updated word representations, compute D PCA components (the dominating D directions): $c_1, ..., c_D = \text{PCA}(\{\tilde{r}(w), w \in V\})$
4. Process the representations eliminating the D dominant directions: $r'(w) = \tilde{r}(w) - \sum_{i=1}^{D}(c_i^\intercal r(w))c_i$

Regarding the dominant directions, D depends on the representations (e.g. their dimension, the training methods) and on the downstream application. Mu et al. (2017) suggest that choosing D around $d/100$ for word representations with dimension d works well across multiple languages, representations and applications. Since our representations have a dimension $d = 300$, we started testing with $D = 3$. After that, we tried to increase the value to $D = 4$ and $D = 10$, obtaining worse results overall. Finally, we tested $D = 2$, but the results were also lower. Considering this, for our model we selected $D = 3$.

3.4 Cross-Lingual Contextual Embeddings

As previously explained, NeuralCoref only includes some simple contextual information about the mention in the features by including the embeddings of some words around it. For our model, we created an additional single mention feature, whose objective was to provide more contextual information through contextual embeddings. For that purpose, we used sentence-transformers library[5], which provides models based on Transformer networks capable of computing contextual sentence representations. Since we were working with two languages, we focused on the multilingual models, namely: *distiluse-base-multilingual-cased*, a multilingual knowledge distilled version of Multilingual Universal Sentence Encoder (mUSE).

Multilingual knowledge distillation (Reimers and Gurevych, 2020) is a method that allows creating multilingual versions from previously monolingual models. It can also be applied to previous multilingual models in order to expand the number of supported languages. Its training is based on the idea that a translated sentence should be mapped to the same location in the vector space as the original sentence (meaning the vector spaces are aligned). The method requires a fixed teacher model M_t, that produces sentence embeddings with the desired properties in one or more source languages. It also requires a set of parallel translated sentences $((s_1, t_1), ..., (s_n, t_n))$, with s corresponding to sentences in one of the source languages and t to sentences in one of the target languages. With these resources a multilingual student model M_s is trained to mimic the teacher one, such that the same sentence is mapped to the same vector by both models: $M_t(s_i) \approx M_s(s_i)$ and

[3] https://github.com/facebookresearch/MUSE

[4] https://fasttext.cc/docs/en/crawl-vectors.html

[5] https://github.com/UKPLab/sentence-transformers

$M_t(t_i) \approx M_s(t_i)$. Mean-Squared Error (MSE) is used to train the model.

For *distiluse-base-multilingual-cased*, the teacher model used was mUSE Yang et al. (2019) and the multilingual student model used was XLM-RoBERTa, pretrained on 100 languages. mUSE computes multilingual sentence embeddings using a dual-encoder Transformer architecture. The Transformer encoder is used to compute context-aware representations of the tokens in a sentence, which are then averaged together to obtain the sentence embedding. It was trained for 16 languages in a multi-task setup, including question-answer prediction, translation ranking and natural language inference.

XLM-RoBERTa was trained to mimic mUSE with parallel data for 50 languages (including Portuguese and Spanish), obtaining the distilled mUSE model.

We used *distiluse-base-multilingual-cased* to compute span embeddings for the new feature. Based on the existent feature containing span vectors, we tried the following variations:

1. Embedding for the mention.
2. Embedding for the span going from the fifth word before the mention to the fifth after.
3. Embeddings for: the mention, the span going from the fifth word before the mention to the fifth after.
4. Embeddings for: the mention, the five words before the mention, the five words after the mention, the sentence and the document.

The option that presented better results was the third new feature variation, so we used it for our model.

3.5 Data Augmentation

As we explained before, one of the challenges for Portuguese co-reference resolution is the shorter amount of annotated data available. To explore a solution for that problem we decided to do data augmentation, translating the Spanish training data to Portuguese and vice-versa, getting more training data for each language. For the translator we used the googletrans library[6]. Algorithm 1 shows our approach for translating each sentence in the documents.

The main idea behind it is that Portuguese and Spanish are similar languages with similar sentence constructions. So, when translating a sentence, the translation should have a similar structure, with the corresponding words in close positions. Step 6 of the algorithm is based on that notion: the match with the closest position is the most likely to be correct, so it is chosen. Following this idea, we defined a threshold of 3 for the position difference. We tested higher values, but the greater the margin, the greater the number of wrong annotations. This threshold is a balance between flexibility and wrong annotations. We made an exception for one-to-one matches (i.e. only one match in the translation and the original sentence), as shown in step 7. We assumed those were right, regardless of their positions. When annotating a match, it is possible that there is already another annotation with the same boundaries. In those cases, only one can be correct, so we compared their corresponding *dist* values and picked the annotation with the smallest.

We tried using only exact matches, but those are a small percentage (less than 50%). One of the reasons behind it is that mentions are translated without their context, generating small differences when compared to its correspondent in the sentence. To allow those small differences, we identified fuzzy_matches using fuzzywuzzy[7] and fuzzysearch[8] libraries.

We used fuzzywuzzy to verify if the fuzzy_match between t_m and t_s was above a score threshold. Fuzzywuzzy measures sequence similarity on a scale from 0 to 100, weighting different algorithms (e.g. SequenceMatcher Ratio, PartialRatio, SortedRatio) and selecting the best score. Since we only wanted very similar cases, we tried setting the threshold to 90. However, very few matches were above it so we lowered it to 80, accepting more matches, the vast majority of them correct.

If the fuzzy_match between t_m and t_s was above the threshold, we used fuzzysearch to find the subsequences in t_s that approximately match t_m. Fuzzysearch uses Levenshtein Distance for that purpose, imposing a threshold for a maximum distance. We set that maximum value to 5, allowing only small changes or typos (e.g. a missing preposition or article).

Algorithm 1 Translation per sentence

```
 1: Translate sentence $s$ - $t_s$
 2: for each mention $m$ in $s$:
 3:     Translate $m$ - $t_m$
 4:     if there are matches of $t_m$ in $t_s$:
 5:         Count the matches - $count$
 6:         Get the match whose $t_m's$ $1^{st}$ word position
              in $t_s$ is closer to $m's$ $1^{st}$ word position in $s$
              and annotate the distance ($dist$)
 7:         if $count = 1$ and ($m$ only appears once in $s$
              or $dist \leq 3$):
 8:             Annotate the match in $t_s$
 9:         else if $count > 1$ and $dist \leq 3$:
10:             Annotate the match in $t_s$
11:     else:
12:         for fuzzy_match of $t_m$ in $t_s$:
13:             Do steps 5-6
14:         if $count > 0$ and $dist \leq 3$:
15:             Annotate the match in $t_s$
```

The data obtained from Spanish translation to Portuguese ended up with 41000 mentions (52% of the original annotations) and the data from Portuguese translation to Spanish with 5314 mentions (59%). This represents a good increase in the amount of data for Portuguese. Although we used more restrictive conditions, the translation algorithm still allows some annotation errors, as expected.

3.6 HyperParameter Choices and Model Training Strategy

Regarding the model hyperparameters, we kept the slack-rescaled costs used in the NeuralCoref ranking loss: 0.8 for a false new, 0.4 for a false link and 1 for a wrong link. We set the initial learning rate for each

[6] https://pypi.org/project/googletrans/

[7] https://pypi.org/project/fuzzywuzzy/

[8] https://pypi.org/project/fuzzysearch/

training phase to 10^{-3}, the minimum learning rate to 10^{-8} and the patience to 5. We kept these values fixed during our training, not tuning them.

We also made some changes in the NeuralCoref training strategy. During our training we used the CoNLL F1-Score on the development set as evaluation metric. For a training strategy analysis, we used the base version of our model without the additional methods (PCA, Distilled mUSE and Data Augmentation), supplied with gold mention boundaries.

We started by changing the criteria to move between the three training phases. We added a call-out function which lowers the learning rate by 10^{-1} if the CoNLL F1-Score has no improvements for more than 5 epochs (patience), until the learning rate reaches the minimum value. Only then, after more than 5 epochs without improving the CoNLL F1-Score with the minimum learning rate, we would go to the next training phase. We did not specify a maximum number of epochs for training or any early stop condition. We used Adam (Kingma and Ba, 2017) as the optimizer for training.

We trained the model with these parameters and criteria, and we noticed that the second phase (Top-pairs) was lowering the development set CoNLL F1-Score instead of improving it, in comparison with the results from the first phase (All-Pairs). Given this information, we tried to eliminate the second training step, working only with the other two, which presented improvements on the CoNLL F1-Score. Analyzing in more detail the epochs, we observed that the best CoNLL F1-Score for each training phase was obtained in their first few epochs, without improving on the following ones. To try to improve the results over the epochs, we implemented a mixed training, switching between All-Pairs and Ranking phases - train 5 epochs with All-Pairs loss, then 5 epochs with Ranking loss, 5 epochs with All-Pairs loss again, and so on - loading the best model at switching and maintaining the learning rate updates as before. This showed mild improvements, again obtained on the early epochs. Finally, we changed the mixed training strategy to switch between All-Pairs and Ranking when the CoNLL F1-Score stops improving, instead of running a specific number of epochs before that. The model trained with this strategy achieved the best results so we used it as the Base model of our proposal.

4 Experimental Evaluation

This section presents the experimental evaluation of the proposed model, detailing the datasets and evaluation metrics used in the experiments, and then discussing the obtained results.

To focus on evaluating our model on the classification task of co-reference resolution, we ran experiments supplying the model with gold mention boundaries covering the two scenarios: monolingual and cross-lingual.

For the monolingual scenario we trained the model on Portuguese and Spanish data separately. For Portuguese we used the Portuguese MUSE embeddings and for Spanish we used the Spanish MUSE embeddings. We tested adding each proposed method to the Base model individually and then combining them. The results are reported on the test portion of each dataset.

For the cross-lingual scenario we trained the model simultaneously with Portuguese and Spanish data. We used both Portuguese and Spanish MUSE embeddings. For words appearing in both files we used the Spanish representation (either one should work since they are aligned). Similarly, we tested adding each proposed method to the Base model individually and then combining them. We reported the results on Portuguese and Spanish test sets individually, to evaluate the performance of the cross-lingual model on each language.

We also ran experiments in both the monolingual and cross-lingual scenarios with our mention detection mechanism instead of gold mention boundaries. The mention detection algorithm generates many candidates, requiring more computational resources, such as time and memory. Since the additional proposed methods further increase the resources needed (e.g. adding the contextual embeddings feature implies bigger feature vectors and data augmentation implies more mentions), we only tested this variant on the Base model.

4.1 Datasets

Similar to the work of Cruz et al. (2018), the datasets used to train and test our model were Corref-PT for Portuguese, and AnCora-CO-ES for Spanish. Corref-PT is the largest Portuguese dataset annotated with co-reference information (Brazilian variant, since European corpora are rarer), containing 124K tokens and 182 documents. Similarly, AnCora-CO-ES is the largest dataset available for Spanish with co-reference information, containing 380K tokens and 1183 documents.

We also did a similar pre-processing of both datasets. The Spanish corpus was already divided, so we used the corresponding train, development and test sets. For the Portuguese corpus, we divided its documents randomly, using 60%, 20% and 20% for the train, development and test sets, respectively.

Additionally, we analyzed some examples in each dataset. We noticed that some names in both datasets were written in a single line with underscores separating their words, instead of having each separate word in a different line (e.g. *Instituto_de_Agronomia*, instead of *Instituto / de / Agronomia*). This affects SpaCy's anottations for those tokens, making it harder to correctly detect those mentions. Additionally, the embeddings for both are different, since the first would use just one word embedding and the second would use the average for the three word embeddings. Having the separate embeddings can help the model detecting similarity with a co-referent mention using one of the words (e.g. a next reference as *o Instituto*). For these reasons, we decided to modify both datasets and separate the words in the underscores without affecting the co-reference annotations, by starting them on the first word and ending on the last one.

We also observed that some annotated gold mentions started and/or ended with a punctuation mark. The mention detection function cleans the candidates, in order for them not to start/end in punctuation (e.g. *a casa* instead of *a casa ,*), so those examples would never be considered as candidate mentions. Since the punctuation does not change the mention, we considered those cases simple errors, so we corrected them in both datasets by changing the mention start (or end) to the next (or previous, respectively) word.

4.2 Evaluation Metrics

To evaluate our model, we reported results on the three most common metrics used for the evaluation of co-reference resolution systems: MUC, B^3 and $\mathrm{CEAF_e}$. We also reported the results on the CoNLL metric, which combines the previous ones. These were the metrics reported in the related works presented, allowing us to make comparisons.

MUC, B^3 and $\mathrm{CEAF_e}$ are defined in terms of how they calculate Precision and Recall. For the three metrics, the F1-Score is computed as a harmonic mean of Precision and Recall. The CoNLL metric is the mean of the three F1-Scores.

Consider $K = k_i : i = 1, 2, ..., |K|$ as the gold entity set and $S = s_i : i = 1, 2, ..., |S|$ as the predicted entity set. k_i and s_i represent the entities (i.e. clusters), with $|K|$ and $|S|$ being the number of mentions.

MUC (Vilain et al., 1995) is a link based metric that operates by comparing the entities defined by the gold links and the predicted links, instead of the links themselves. Recall (or Precision) is based on the minimum number of links that need to be added to the predicted entities (or gold entities, respectively), in order to get them aligned with the gold ones (or predicted ones, respectively). The following equations show how to compute Precision and Recall:

$$\text{Precision} = \frac{\sum_{s_i \in S}(|s_i| - |p(s_i)|)}{\sum_{s_i \in S}(|s_i| - 1)} \tag{1}$$

$$\text{Recall} = \frac{\sum_{k_i \in K}(|k_i| - |p(k_i)|)}{\sum_{k_i \in K}(|k_i| - 1)} \tag{2}$$

In Equation 2, $p(k_i)$ is a partition of k_i containing subsets of it, where each subset is created by intersecting k_i with the predicted entities that overlap with it. Similarly, in Equation 1, $p(s_i)$ is a partition of s_i relative to the gold standard.

B^3 (Bagga and Baldwin, 1998) measures performance on the mention level. It computes individual Precision and Recall for each mention and then computes the average of these values to get the final Precision and Recall, as shown in the following equations:

$$\begin{aligned} \text{Precision}(m_i) &= \frac{|k_{m_i} \cap s_{m_i}|}{|s_{m_i}|} \\ \text{Precision} &= \frac{\sum_i^{M_s} \text{Precision}(m_i)}{M_s} \end{aligned} \tag{3}$$

$$\begin{aligned} \text{Recall}(m_i) &= \frac{|k_{m_i} \cap s_{m_i}|}{|k_{m_i}|} \\ \text{Recall} &= \frac{\sum_i^{M_k} \text{Recall}(m_i)}{M_k} \end{aligned} \tag{4}$$

For each mention m_i, Recall (or Precision) computes the number of correct mentions in the predicted entity containing m_i over the number of mentions in the gold entity (or predicted entity, respectively) containing m_i. In Equations 3 and 4, s_{m_i} and k_{m_i} represent the predicted and gold entities containing m_i, respectively. When computing the final measures, M_s and M_k are the numbers of predicted and gold mentions, respectively. Using gold mention boundaries, $M_s = M_k$.

However, when using a mention detection system, there are *twinless mentions* - predicted mentions that are not mapped to any gold mention and vice-versa. To overcome that problem, when performing mention detection the following modifications were incorporated in the scorer (Cai and Strube, 2010):

- Include the non-detected gold mentions in the prediction as singletons.
- Discard the detected mentions not included in the gold ones and resolved as singletons.
- Computing Recall: discard the *twinless predicted mentions* in the predicted mentions set.
- Computing Precision: add the *twinless predicted mentions* to the gold mentions set as singletons.

CEAF (Luo, 2005) computes the alignment between gold and predicted entities. It finds the best one-to-one mapping between gold and predicted entities, using a similarity measure (ϕ) for each pair of entities to determine the value of each possible alignment. Every predicted entity is aligned with at most one gold entity. The best mapping function g^* (i.e. the one with the highest total similarity) is used to compute Precision and Recall, as shown in Equations 5 and 6. K^* is the set of gold entities included in the best mapping.

$$\text{Precision} = \frac{\sum_{k_i \in K^*} \phi(k_i, g^*(k_i))}{\sum_{s_i \in S} \phi(s_i, s_i)} \tag{5}$$

$$\text{Recall} = \frac{\sum_{k_i \in K^*} \phi(k_i, g^*(k_i))}{\sum_{k_i \in K} \phi(k_i, k_i)} \tag{6}$$

We used the entity based variant, $\mathrm{CEAF_e}$, which computes the similarity as the relative number of common mentions between the two entities: $\phi(k, s) = \frac{2 \cdot |k \cap s|}{|k| + |s|}$. With $\mathrm{CEAF_e}$, the denominator of Equations 5 and 6 corresponds to the number of predicted and gold entities, respectively.

4.3 Results

Table 2 reports the results of the proposed model trained on a monolingual scenario for each language, using gold mention boundaries.

Regarding the monolingual training on Spanish data, the models combining Base+DmUSE and Base+PCA+DmUSE+DA had the best performances, with 74.7% and 74.6% CoNLL F1-Scores, respectively. However, each individual method generated improvements over the Base version. The overall results reported on B^3 and $\mathrm{CEAF_e}$ are considerably high, which means there is a good alignment between the predicted and gold entities and that a good percentage of mentions are being resolved to the right entity. MUC results are a little lower, which we assume is related to it disregarding singletons, which are annotated in Spanish data and represent a considerable amount of mentions, implying their correct prediction improves the other metrics. Our model has the best performance, in comparison with the

		MUC			B^3			$CEAF_e$			CoNLL
		P	R	F1	P	R	F1	P	R	F1	Avg F1
ES	Base	62.6	55.4	58.8	82.3	76.6	79.4	80.5	86.2	83.2	73.8
	+PCA	**65.8**	54.9	59.9	**83.5**	77.0	80.1	79.6	**87.6**	83.4	74.5
	+DmUSE	64.1	56.6	60.1	82.8	77.4	80.0	**81.1**	86.9	**83.9**	**74.7**
	+DA	64.1	56.4	60.0	81.5	77.4	79.4	80.6	86.5	83.5	74.3
	+PCA+DmUSE	62.7	56.4	59.4	82.2	77.1	79.6	81.0	86.0	83.4	74.1
	+PCA+DA	62.0	57.0	59.4	81.6	77.4	79.4	80.9	84.8	82.8	73.9
	+DmUSE+DA	63.0	56.8	59.7	82.3	77.4	79.8	**81.1**	86.0	83.4	74.3
	+PCA+DmUSE+DA	63.6	**57.4**	**60.3**	82.6	**77.9**	**80.2**	80.9	85.7	83.2	74.6
PT	Base	84.6	57.9	68.7	**87.8**	58.3	70.1	38.9	72.3	50.6	63.1
	+PCA	77.4	60.9	68.1	79.5	59.8	68.2	44.9	70.9	**55.0**	63.9
	+DmUSE	76.0	61.9	68.2	76.3	60.7	67.6	**45.7**	68.8	54.9	63.6
	+DA	77.5	**62.4**	69.1	74.8	**60.9**	67.2	43.9	67.2	53.1	63.1
	+PCA+DmUSE	79.9	60.3	68.7	81.9	60.0	69.3	42.4	70.8	53.0	63.7
	+PCA+DA	83.3	59.0	69.1	85.8	59.4	70.2	40.7	**72.9**	52.2	63.9
	+DmUSE+DA	**85.3**	59.3	70.0	87.0	59.4	**70.6**	39.8	**72.9**	51.5	**64.1**
	+PCA+DmUSE+DA	83.6	59.1	**69.2**	86.6	59.1	70.2	40.5	72.7	52.0	63.8

Table 2: Evaluation results for monolingual co-reference resolution on AnCora-CO-ES and Corref-PT datasets using gold mention boundaries.

previous works' results reported in Table 1 (+4.8% on CoNLL F1-Score).

For the monolingual training on Portuguese data, the model combining Base+DmUSE+DA had the best performance, with 64.1% CoNLL F1-Score. Individually, DA was the only method that did not improve the model; however, it improved the percentage of links and mentions predicted correctly from the gold data (higher Recall MUC and B^3 values). Results are lower compared to the ones obtained for Spanish data, as expected. That happens because Portuguese is a less resourced language. Despite that, we achieved promising results, reporting a better performance in comparison with the previous models' results presented in Table 1 (+14.4% on CoNLL F1-Score).

In particular, we think that DmUSE improves the results on both languages not only due to the additional contextual information, but also due to providing a representation for out-of-vocabulary words existent in the test but not in the training data. Those words had no MUSE embeddings so previously they were represented with arrays of zeros.

Table 3 reports the results of the proposed cross-lingual model trained simultaneously with both Spanish and Portuguese data, using gold mention boundaries, and tested on each individual test set. The cross-lingual model succeeded in generalizing and performing co-reference resolution for both languages, offering competitive results for both Portuguese and Spanish. However, when compared to the monolingual approach, we cannot say that this generalization achieves better performance for either language. The only previous work exploring cross-lingual settings was the one presented by Cruz et al. (2018). They only reported results for the Portuguese test set, as they focused of direct transfer learning from Spanish to Portuguese. Our approach achieved better results in comparison with theirs (+18.9% on CoNLL F1-Score), which may be partially related to Portuguese data also being used for training.

On this cross-lingual training scenario, the model combining Base+DmUSE+DA had the best performance in the Spanish test data, with 74.9% CoNLL F1-Score. For the Portuguese test data, the best performance was achieved by the model combining Base+PCA+DmUSE, with 63.7% CoNLL F1-Score.

Tables 4 and 5 report the results for the Base model trained in the monolingual and cross-lingual scenarios, respectively, but using our mention detection function instead of gold mention boundaries. Regarding the mentions detected, the Recall values are high for both languages, around 85%, meaning 85% of the gold mentions were successfully detected. However, the Precision values are lower, meaning a lot of incorrect mentions were detected. For Spanish 48.3% of the mentions detected were right, while for Portuguese it was only 16.7%.

As expected, the results for both scenarios are lower when compared to those obtained using gold mentions. Flaws in mention detection provoke error propagation to the co-reference resolution task, affecting its metrics. Similarly to the results obtained with the gold mentions, the cross-lingual model is capable of generalizing but reports slightly lower results in comparison with the corresponding monolingual models.

5 Conclusions and Future Work

In this paper we presented a state-of-the-art neural co-reference resolution model for Portuguese and Spanish data, built on a previous one developed for English data - NeuralCoref. We tested additional methods to improve the model, namely post-processing for cross-lingual word embeddings and adding a mention feature based on cross-lingual contextual embeddings. Exploring another solution for the smaller amount of Portuguese resources for this task, we translated each training set to the other language, generating more co-reference annotated data.

Our model was trained on a monolingual scenario for each language, using gold mention boundaries, and achieved a better performance in comparison with the existing ones, trained and tested on the AnCora-CO-ES and Corref-PT corpora. We also presented a cross-

		MUC			B^3			$CEAF_e$			CoNLL
		P	R	F1	P	R	F1	P	R	F1	Avg F1
ES	Base	62.9	56.6	59.6	81.3	76.8	79.0	81.1	86.0	83.5	74.0
	+PCA	62.6	54.8	58.5	82.1	76.4	79.1	80.3	86.4	83.2	73.6
	+DmUSE	63.3	**56.7**	59.8	80.6	77.1	78.8	**81.5**	86.7	84.0	74.2
	+DA	63.6	56.1	59.6	83.1	77.0	80.0	80.8	86.6	83.6	74.4
	+PCA+DmUSE	65.5	56.2	60.5	83.0	77.2	80.0	80.7	**87.7**	**84.1**	73.9
	+PCA+DA	62.4	55.2	58.6	81.7	76.8	79.1	80.5	86.1	83.2	73.6
	+DmUSE+DA	**66.7**	56.6	**61.2**	83.0	**77.4**	80.1	79.8	87.2	83.3	**74.9**
	+PCA+DmUSE+DA	65.6	55.1	59.9	**84.9**	76.4	**80.4**	79.4	87.1	83.1	74.5
PT	Base	73.9	**64.0**	**68.6**	68.8	**62.7**	65.6	46.6	63.6	53.8	62.7
	+PCA	73.2	62.7	67.5	71.8	61.8	66.5	47.6	66.3	55.4	63.1
	+DmUSE	75.1	62.1	68.0	73.3	60.9	66.5	45.6	67.0	54.3	62.9
	+DA	**82.5**	55.8	66.6	**87.0**	56.6	68.5	38.5	**72.4**	50.3	61.8
	+PCA+DmUSE	74.4	63.3	**68.4**	72.7	62.2	67.0	**47.7**	67.1	**55.8**	**63.7**
	+PCA+DA	77.0	59.1	66.9	78.6	58.8	67.2	42.1	68.8	52.3	62.1
	+DmUSE+DA	80.8	58.5	67.9	81.8	58.5	68.2	39.8	69.8	50.7	62.3
	+PCA+DmUSE+DA	79.9	59.7	68.3	81.2	59.7	**68.8**	41.9	70.7	52.6	63.3

Table 3: Evaluation results for cross-lingual co-reference resolution on AnCora-CO-ES and Corref-PT datasets using gold mention boundaries.

		Mentions			MUC			B^3			$CEAF_e$			CoNLL
		P	R	F1	P	R	F1	P	R	F1	P	R	F1	Avg F1
ES	Base	48.3	85.3	61.7	64.8	43.3	51.9	42.9	63.3	51.2	31.7	77.2	45.0	49.4
PT	Base	16.7	84.8	28.8	60.7	31.0	41.0	15.5	37.7	22.0	3.6	62.4	6.8	23.3

Table 4: Evaluation results for monolingual co-reference resolution on AnCora-CO-ES and Corref-PT datasets using mention detection.

		Mentions			MUC			B^3			$CEAF_e$			CoNLL
		P	R	F1	P	R	F1	P	R	F1	P	R	F1	Avg F1
ES	Base	48.3	85.3	61.7	65.7	42.2	51.4	43.3	62.9	51.3	31.7	77.5	45.0	49.2
PT	Base	16.7	84.8	28.8	60.9	28.7	39.0	15.6	36.5	21.9	3.5	61.7	6.6	22.5

Table 5: Evaluation results for cross-lingual co-reference resolution on AnCora-CO-ES and Corref-PT datasets using mention detection.

lingual variant of the co-reference resolution model, trained simultaneously on data from both languages. To the best of our knowledge, it is one of the first systems exploring cross-lingual learning with Spanish and Portuguese. The model reports competitive results, in comparison with in-language trained models. Additionally, we developed a mention detection function, capable of identifying candidate mentions on the AnCora-CO-ES and Corref-PT corpora, running experiments with it. The results obtained were considerably lower in comparison with the models using gold mentions, confirming that errors in mention detection affect the co-reference resolution task.

Despite the interesting results, there is room for improvement in future work. As the model hyperparameters were not tuned, a simple future improvement would be to fine tune them. In the training process, the number of positive and negative (i.e. co-referent and non co-referent) instances is highly unbalanced towards the positive side, for both languages. Hence, a possible route is to explore undersampling, as proposed by Cruz et al. (2018). Regarding the word embeddings, one possible option would be to replace the FastText embeddings by cross-lingual contextual embeddings such as ELMo or BERT, using only contextual representations. There are some methods available for their alignment, like the one presented by Schuster et al. (2019). Alternatively there are also multilingual models already available, similar to the distilled mUSE. Focusing on the model, a new feature could be added.

For a mention-pair, the contextual representations of its mentions (obtained from distilled mUSE) can be compared using cosine similarity, and that value can be added as a new mention-pair feature.

Another promising line of work is to improve the mention detection. Following recent research, the model can be adapted to jointly tackle mention detection and co-reference resolution, so those tasks share the same optimization goal, as in the work developed by Lee et al. (2017). We believe that an improvement to mention detection subtask would be reflected as an improvement on the co-reference resolution performance.

References

Attardi, G., Simi, M., and Dei Rossi, S. (2010). TANL-1: Coreference Resolution by Parse Analysis and Similarity Clustering. In *Proceedings of the International Workshop on Semantic Evaluation*, pages 108–111. Association for Computational Linguistics.

Bagga, A. and Baldwin, B. (1998). Algorithms for Scoring Coreference Chains. In *Proceedings of the International Conference on Language Resources and Evaluation Workshop on Linguistics Coreference*, pages 563–566. Granada.

Cai, J. and Strube, M. (2010). Evaluation Metrics for End-to-End Coreference Resolution Systems. In *Proceedings of the SIGDIAL 2010 Conference*, pages 28–36. Association for Computational Linguistics.

Clark, K. and Manning, C. D. (2016). Improving Coreference Resolution by Learning Entity-Level Distributed Representations. *arXiv:1606.01323*.

Cruz, A. F., Rocha, G., and Cardoso, H. L. (2018). Exploring Spanish Corpora for Portuguese Coreference Resolution. In *Proceedings of the International Conference on Social Networks Analysis, Management and Security (SNAMS)*, pages 290–295. IEEE.

Fonseca, E., Sesti, V., Antonitsch, A., Vanin, A., and Vieira, R. (2017a). CORP: Uma Abordagem baseada em Regras e Conhecimento Semântico para a Resoluçao de Correferências. *Linguamática*, 9(1):3–18.

Fonseca, E., Sesti, V., Collovini, S., Vieira, R., Leal, A., and Quaresma, P. (2017b). Collective Elaboration of a Coreference Annotated Corpus for Portuguese Texts. In *Proceedings of Workshop on Evaluation of Human Language Technologies for Iberian Languages (IberEval)*, volume 1881, pages 68–83. CEUR-WS.

Fonseca, E., Vanin, A., and Vieira, R. (2018). Mention Clustering to Improve Portuguese Semantic Coreference Resolution. In *Proceedings of the International Conference on Applications of Natural Language to Information Systems*, pages 256–263. Springer.

Grave, E., Bojanowski, P., Gupta, P., Joulin, A., and Mikolov, T. (2018). Learning Word Vectors for 157 Languages. *arXiv:1802.06893*.

Kingma, D. P. and Ba, J. (2017). Adam: A Method for Stochastic Optimization. *arXiv:1412.6980*.

Kobdani, H. and Schütze, H. (2010). SUCRE: A Modular System for Coreference Resolution. In *Proceedings of the International Workshop on Semantic Evaluation*, pages 92–95. Association for Computational Linguistics.

Lee, K., He, L., Lewis, M., and Zettlemoyer, L. (2017). End-to-end Neural Coreference Resolution. *arXiv:1707.07045*.

Luo, X. (2005). On Coreference Resolution Performance Metrics. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 25–32. Association for Computational Linguistics.

Mu, J., Bhat, S., and Viswanath, P. (2017). All-but-the-Top: Simple and Effective Postprocessing for Word Representations. *arXiv:1702.01417*.

Oliveira, H. G. (2012). *Onto.PT: Towards the Automatic Construction of a Lexical Ontology for Portuguese.* PhD thesis, University of Coimbra/Faculty of Science and Technology.

Recasens, M., Màrquez, L., Sapena, E., Martí, M. A., Taulé, M., Hoste, V., Poesio, M., and Versley, Y. (2010). SemEval-2010 Task 1: Coreference Resolution in Multiple Languages. In *Proceedings of the International Workshop on Semantic Evaluation*, pages 1–8. Association for Computational Linguistics.

Recasens, M. and Martí, M. A. (2010). AnCora-CO: Coreferentially annotated corpora for Spanish and Catalan. *Language resources and evaluation*, 44(4):315–345.

Reimers, N. and Gurevych, I. (2020). Making Monolingual Sentence Embeddings Multilingual using Knowledge Distillation. *arXiv:2004.09813*.

Sapena, E., Padró, L., and Turmo, J. (2010). RelaxCor: A Global Relaxation Labeling Approach to Coreference Resolution. In *Proceedings of the International Workshop on Semantic Evaluation*, pages 88–91. Association for Computational Linguistics.

Schuster, T., Ram, O., Barzilay, R., and Globerson, A. (2019). Cross-lingual Alignment of Contextual Word Embeddings, with Applications to Zero-shot Dependency Parsing. *arXiv:1902.09492*.

Silva, W. D. C. (2013). *Aprimorando o Corretor Gramatical CoGrOO.* PhD thesis, University of São Paulo.

Stylianou, N. and Vlahavas, I. (2019). A Neural Entity Coreference Resolution Review. *arXiv:1910.09329*.

Sukthanker, R., Poria, S., Cambria, E., and Thirunavukarasu, R. (2018). Anaphora and Coreference Resolution: A Review. *arXiv:1805.11824*.

Vilain, M., Burger, J., Aberdeen, J., Connolly, D., and Hirschman, L. (1995). A Model-Theoretic Coreference Scoring Scheme. In *Proceedings of the Conference on Message Understanding*, pages 45–52. Association for Computational Linguistics.

Yang, Y., Cer, D., Ahmad, A., Guo, M., Law, J., Constant, N., Abrego, G. H., Yuan, S., Tar, C., Sung, Y.-H., et al. (2019). Multilingual Universal Sentence Encoder for Semantic Retrieval. *arXiv:1907.04307*.