Portuguese Verb Sense Disambiguation Using Parallel Corpus

Valentyn Hulevych
Instituto Superior Técnico
Lisbon, Portugal
valentyn.hulevych@tecnico.ulisboa.pt

ABSTRACT

Semantic ambiguity is a very frequent linguistic phenomenon in texts and it has a strong impact in many tasks pertaining to various fields of Natural Language Processing (NLP), such as Machine Translation, where it is important to identify the specific meaning of a word based on its context. This process is called Word Sense Disambiguation (WSD). Normally, large quantities of annotated text are required to build the training *corpus* used in statistically-based WSD systems. Unfortunately, these *corpora* are scarce for many languages and their production is expensive and very time-consuming.

This work presents an approach to semi-automatically generate a training *corpus* annotated with the meanings of the verbs, to be used with Machine Learning (ML) techniques in the WSD task. For this purpose, the Portuguese processing chain STRING is used. The annotation of training *corpus* is based on the disambiguation resulting from *corpora* of parallel texts in various pairs of languages. This document describes the generation of new training files produced within the scope of a ML approach to WSD. This approach is applied to a small range of verbs, whose ambiguity has a strong impact on the global ambiguity of the reference corpus.

CCS CONCEPTS

• **Computing methodologies** → **Natural language processing**; **Machine learning**;

KEYWORDS

Natural Language Processing; Verb Sense Disambiguation; Multilingual Parallel corpora

1 INTRODUCTION

NLP is a field of Artificial Intelligence (AI) that deals with semantic ambiguity. It is one of the phenomena that most difficult the process of automatic comprehension of natural languages. It corresponds to situations where the same lexical item have more then one meaning. Consider the following examples: *O Pedro conta as moedas antes de sair.* (Peter counts the coins before leaving.); *O Pedro conta contigo para o ajudares.* (Peter count/relies on you to help him.). The verb *contar* 'to count' can have multiple meanings in different contexts and, hence, it can be translated by different English words. In the first sentence the verb *contar* translates into 'count' and in the second into 'count' or 'rely on'.

The process of identifying the specific meaning of a word based on its context is called Word Sense Disambiguation (WSD) [8]. WSD is an important task in fields such as Machine Translation (MT), Information Retrieval (IR) or content characterization.

The main goal of this work is to get better results in the Verb Sense Disambiguation (VSD) task on STRING [7], a statistical and rule-based NLP system developed at Spoken Language Systems Laboratory (L2F). As the verb is arguably one of the most important elements in determining the sense of the words in a sentence, many experiments have been made to correctly identify its sense when used in a context.

Currently, there are two main approaches that are used by STRING to address the WSD task. The first one is a rule-based approach [10, 13], that only uses the ViPEr database [2, 3]. This database provides the syntactic and semantic information about the verbs' lexico-grammatical constructions that enable the system to generate disambiguation rules. These disambiguation rules are, then, adequately reordered to perform disambiguation.

The other one is a ML approach, that makes use of manually annotated *corpus*, where verbs have been tagged for their senses according to ViPEr verb classes. In this way, it is possible to produce a model that automatically classifies the meaning of verbs. Different types of ML algorithms were tested [12] and the most accurate one (Naive Bayes) is used.

The goal of this work is to apply the ML approach on more verbs and with a larger training *corpus*. Parallel *corpora* that where manually translated on multiples languages allow the use of translations of texts into other languages to help disambiguating some words [11]. If a word has different meanings and a given translation stands for one of them in a regular way, then it should be possible to tag that instance with a specific word sense and discard the other potential meanings of that word.

The idea underlying this study is to use a parallel multilingual *corpus* for sense tagging the targeted verbs or, at least, to produce a pre-annotated *corpus* in order to facilitate and speed up the annotation process.

2 TAGGING SENTENCES USING PARALLEL CORPORA

As mentioned before, the main goal of this work is to apply the ML approach over more verbs, generating more training material. A new module responsible for the generation of tagged data with multilingual tuples was implemented. This module resorts to a sentence-aligned multilingual *corpora* and on tables that map words between different languages.

2.1 Corpus Alignment

To obtain the sentence-aligned multilingual *corpora*, the Europarl *corpus* [5] was used. This *corpus* is composed of the proceedings of European Parliament, for more then 10 years, and translated into more then 20 languages. These characteristics maximize the

quantity of data that can be generated, as well as the variety of language pairs that can be explored.

The process of alignment starts with some pairwise pre-processing steps. It includes tokenization, removal of empty lines and XML tags, as well as alignment, at sentence level, of two paragraph-aligned texts. The pair of generated aligned sentences is composed by a Pivotal Language (PL) file, which refers to a primary language, and by a foreign language file, which will be used for disambiguation. For this project, Portuguese was used as PL and 5 other foreign languages were used for the alignments: English, Spanish, French, Italian and German. These languages were chosen considering the number of sentences present in the *corpus* for each language, thus reducing the loss of information in the alignment and maximizing the amount of available data. After this pre-processing stage, 5 different, sentence-aligned bitexts were produced. As it cannot be know beforehand which language combination would produce better results in the disambiguation process, the previously aligned bitexts were pooled together in different combinations.

For that, the PL was used as a bridge. When two bitexts, involving 2 different language pairs, are aligned, there are sentences in one of the bitexts that cannot be mapped onto any sentence of the other bitext, and must therefore be removed. For example, from the two bitexts Portuguese-English and Portuguese-French, the multilingual aligned texts Portuguese-English-French were produced. From an original set of Portuguese-English sentences and of a number of Portuguese-French sentences, only a certain number of sentences could be aligned for the three languages.

To align the previously produced bitexts, an algorithm was developed. When aligning N bitexts, the first bitext is aligned with the second, then these are aligned with the third, and so on, until all N bitexts are aligned with each other. On each iteration, the unique lines from each of bitexts are stored and then removed.

Number of Bitexts	Average Number of Sentences
1	1,756,840
2	1,438,415
3	1,276,940
4	1,095,011
5	971,886

Table 1: *Corpus* size according to the number of aligned bitexts.

Table 1 shows how the average number of sentences decays with the number of aligned bitexts. The averages were calculated based on all possible combinations of previously presented bitexts. When using only one bitext, the *corpus* has almost 1,8 million sentences. However, by adding more bitexts in the alignment process, the *corpus* is reduced to about 200 thousand sentences per set of bitexts. This shows that the number of languages involved in the tags must be limited and that using a language that has fewer sentences can reduce significantly the available data.

2.2 Generation of Word Tables

The second step of data preparation is the generation of the word mapping tables. Even having sentences translated in many languages, they are useless if there is no information about which words correspond to other words in their translation. To overcome this issue, we used GIZA++ [9], which is able to create a mapping between the words of two (sentence-aligned) texts. As GIZA++ models require that all training data be in lowercase, the sentence-aligned *corpus* produced in the previous step was converted into lowercase format.

```
break the|quebrar as|0.5454 0.2029 0.0223 0.0052|0-0 1-1
break the|quebrar o|0.5858 0.2542 0.2156 0.1852|0-0 1-1
break the|se quebrar o|1 0.1271 0.0074 0.0003|0-0 0-1 1-2
break the|quebrar os|0.2222 0.2085 0.0074 0.00710-0 1-1
break their|quebrar o seu|1 0.0471 0.375 0.0008|0-0 1-1 1-2
```

Figure 1: Mapping examples for *quebrar/break*.

As result, for each pair of languages, the bitext composed by the Portuguese *corpus* and a *corpus* in a foreign language produced a translation table, with the mappings between words in the aligned sentences. Consider Figure 1, which exemplifies some mappings for the word *quebrar* 'break', extracted from the Portuguese-English word table produced by GIZA++. The first column contains the phrase translated into the target language, while the second contains the phrase in the source language (here, the PL, Portuguese). The third entry is composed by four values:

- Inverse Phrase Translation Probability (IPTP): $\phi(f|p)$;
- Inverse Lexical Weighting (ILW): $lex(f|p)$;
- Direct Phrase Translation Probability (DPTP): $\phi(p|f)$;
- Direct Lexical Weighting (DLW): $lex(p|f)$;

The IPTP represents the probability of a phrase occurring in the foreign language *corpus* given the Portuguese phrase, while the ILW is the weight associated to that relation, which is calculated based on the applied translation models. The last two metrics are similar, measuring the probability and weight of a Portuguese phrase given a foreign language phrase. In the context of this project, only the first two entries, this is, IPTP and ILW, are relevant, as they map Portuguese words to the corresponding translations.

Finally, the last column of the table indicates the mapping between words. Analyzing the expression *quebrar o seu*, it maps to *break their* with a probability of 1 and a wight of 0.0471. The 0-0 indicates that the word on the index 0 of the first column maps to the word on the index 0 of the second column, meaning that there is a correspondence between *quebrar* and *break*.

2.3 Tagging Module

The Tagging Module is responsible for tagging the sentences in the PL with multilingual tuples. Figure 2 describes the developed architecture. It uses as input the sentence-aligned files produced as described in the Section 2.1, the tables generated by GIZA++ and described in the Section 2.2, and a list with all the inflected forms for a target verb.

The module starts by extracting all sentences in the PL that contain a target verb inflected forms. Next, for each text in the foreign languages, all possible translations for the inflected forms of the target verb are extracted. In this process, the GIZA++ tables (described in Section 2.2) were used. Then, by looking at the parallel sentences, the module searches for one of the possible translations previously

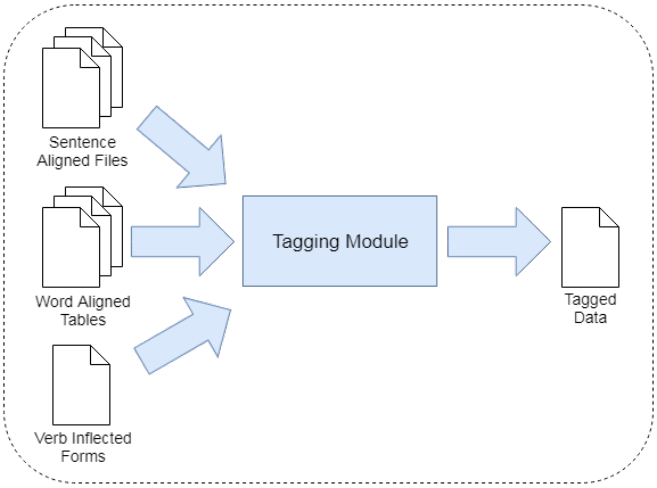

Figure 2: Tagging module architecture.

extracted from the table. In this way, for each original sentence, the corresponding *tag* or *tuple* will be added, which is composed by the translations of the PL word into one or more different languages. It is important to mention that the sentences in the foreign languages were lematized to maximally reduce the multiple forms of the same verb. For that purpose, a Python lemmatizing module named spaCy[1] was used. As there is always an error associated to the lemmatization process, a list with inflexions from STRING is used to mitigate it, at least on the PL sentences.

At the end, the previously described tags will be used as proxy for sense tags in the VSD process. The task of analyzing whether a given tag is able to disambiguate the existent ViPEr classes requires semantic knowledge that, at this stage, can only be provided by a human. For this reason, the mapping of the tags onto the ViPEr classes, a necessary step to produce the annotated *corpus* for STRING ML algorithm, must be done manualy.

3 PILOT STUDY

To elicit insights and draw tentative conclusions about the variables and values to be used on the tagging process and the selection and mapping of tags onto ViPEr classes, some pilot experiments were explored.

The first experiment consisted in exploring the influence of the IPTP and ILW values, presented in Section 2.2. If these two values are too low, the tagging process will be significantly affected by alignment errors. It is possible that a verb maps onto prepositions or auxiliary verbs, making it impossible to get a correct identification of the translations on the parallel sentences. The occurrence of prepositions or auxiliary verbs can be frequent in the *corpus*, meaning that this type of mapping could significantly affect the results. To eliminate these cases, the IPTP and ILW values were successively adjusted, starting with 0, and they were increased until at least the prepositions were eliminated from the tags. It was concluded that establishing a minimal value of 0.3 for IPTP and of 0.001 for ILW discards almost all the errors. Finally, as expected, the lematization process also helped on the elimination of word classes that are not verbs, and it contributed, apparently, to enhance the quality of mappings between verbs of two different languages.

The second experiment consisted in finding out the languages' combinations that produce the best results. For that purpose, 5 verbs were chosen: *saber* 'know', *ver* 'see', *falar* 'speak', *pensar* 'think' and *explicar* 'explain'. These verbs are the five most frequent verbs that had already been annotated and used on the ML disambiguation task. All possible alignments were first generated and all of them were processed by the Tagging Module. The main reason for not using all languages together is related to the fact that the number of aligned sentences in the *corpus* is reduced by about 200 thousand sentences for each extra language aligned. Also, if there is at least one language for which no verb translation was found in the translated sentence, then the sentence on the PL is discarded. There is no way to find the translations in this case, as they can be too distant from the original expression. To complicate matters even more, the texts in the Europarl *corpus* do not have the information about the language in which the original text was produced, meaning that it is not possible to know whether there is any significant influence of the translators native language on the texts' translation process.

Number of languages	Number of sentences	Number of tags
1	4,256	8
2	1,947	28
3	1,154	45
4	698	52
5	458	51

Table 2: Dimension of sentences and different tags by the number of languages aligned with Portuguese.

In Table 2, the average values for all languages' combinations and for the 5 verbs were calculated. The first line means that aligning Portuguese with one language, produces in average 4,256 sentences and 8 different tags. In this case, as there is only one language involved, the tag is composed exclusively by the verb and its translation. The table shows that the number of instances decays steeply with the number of aligned languages, and that the number of tags tends to expand.

We conclude, then, that using only one or at most two languages can produce the best results. If a given language is not able to distinguish the verb meanings by different translations, the probability of distinguishing them by combining more languages is significantly reduced.

4 MAPPING SENTENCES ONTO VIPER CLASSES

When the tagged data was generated, the tags were manually mapped onto ViPEr classes. The task of analyzing whether a given tag was able to disambiguate the existent ViPEr classes required semantic knowledge that, at this stage, could only be provided by a human. For the first experiments, the previous 5 verbs was chosen. These verbs were the five most frequent verbs that had already

[1] https://spacy.io/

been annotated and used on the ML disambiguation task. As it is not possible to always map directly a tag onto the corresponding class, different scripts were developed. These scripts classify sentences according to the context of target verbs. This means that this method is only capable of capturing a subset of instances from each construction. The following lines describe the results of the mapping process for the set of 5 verbs chosen.

- *saber/know:* discarding the rare constructions, the verb *saber* has been attributed to 3 different ViPEr classes corresponding approximately to 3 different word senses. First, this verb can refer to the sensorial experience and be equivalent to the sense of *taste* (class 33) or denote a mental perception and be equivalent to the verb *know* (classes 06 and 34). The difference between these last two constructions is mostly syntactic. In class 06, the verb requires a subclause as its direct object, while in class 34 the verb does not allow a subclause as its complement, and this is introduced by the preposition *de*. Consider Example 1, which clarifies these differences.

  [33] *O iogurte sabe a ananás* (The yogurt tastes like pineapple)
  [06] *O Pedro sabe que a Ana é uma pessoa competente* (Peter knows that Ana is a competent person)
  [34] *Ele não queria saber das chaves* (He does not want to know about the keys)
  Example 1. Different constructions of the verb *saber* 'know' considered in ViPEr.

  The verb construction of class 33 is absent from the *corpus*, meaning that it is not possible to train a model to identify this class. On the other hand, the annotated training file currently used in ML approach had only one sentence with class 33. This is a recurring problem, which is related to the nature of the *corpus*. Regarding the instances of classes 06 and 34, it is not possible to distinguish their syntactic differences by looking at translations alone.
  Nonetheless, two scripts to separate these two constructions were created. To maximize the number of extracted sentences, the Portuguese-English pair and the most frequent tag *<saber, know>* were used. If the verb was followed by the words/phrases *que*, *se*, *de onde*, *de quem*, *de que*, *de quê*, *do que* or *do quê* then it was classified as an instance of class 06; otherwise, when the verb was followed by *de*, *do*, *da*, *dele*, *dela* or *disso* and it was not followed by a subclause, then it was classified as a instance of class 34.
- *ver/see:* the verb *ver* has 2, frequently occurring, ViPEr constructions. The first one is in class 06 and it refers to a process of mental perception. The second one is in class 32C and it refers to visual perception. Consider Example 2, which differentiates the two constructions described above.

  [06] *O Pedro viu que o concurso tinha sido fraudulento* (Peter saw that the contest had been fraudulent)
  [32C] *O Pedro viu um filme* (Peter saw a movie)
  Example 2. Different constructions of the verb *ver* 'see' considered in ViPEr.

  Instances of the construction from class 32C were difficult to identify by translations. Only the combination of Portuguese with Spanish and Italian had a candidate tag, the tuple *<ver, ver, vistare>*. For the construction from class 06, the combination of Portuguese with French and the tags *<ver, constater>* and *<ver, considérer>* were used. Even if it was only possible to correctly distinguish the class 06 construction, this would help to find the occurrences of construction from class 32C, which would be searched in the remaining *corpus*.
- *falar/talk:* at the onset of this analysis, the verb *falar* had been assigned to 3 frequently occurring ViPEr classes, which could be organized into two semantic groups. The first group was composed by the construction from class 32R, and it refers to the ability of speaking a given language. The second semantic group was composed by the constructions from classes 41 and 42S, which refer to a process of communication. Example 3 tries to clarify all the word senses attributed to this verb.

  [32R] *O Pedro fala várias línguas* (Peter speaks several languages)
  [35R] *O Pedro falou em inglês* (Peter spoke in English)
  [35S] *O Pedro não fala com o João.* (Peter does not speak with João)
  [41] *O Pedro falou sobre esse assunto ao João* (Peter spoke about that subject to João)
  [42S] *O Pedro falou com o João em fazer isso* (Peter spoke with João about doing that)
  Example 3. Different constructions of the verb *falar* 'speak' considered in ViPEr.

  The sentences annotated with class 41 refer to an oriented communication process that is asymmetric and where an active speaker (agent) and and passive listener (patient/addressee) are present. The use classified as 42S refers to a non-oriented communication process, which is symmetric and where both participants are agents. Because of this symmetry [1], the NP's denoting the two participants can be coordinated in the subject syntactic slot and a facultative *echo complement*, indicating reciprocity [4], can be added.
  After the analysis of the paired sentences, two new constructions were discovered, which had not been considered yet in ViPEr, as the previously annotated *corpus* did not contain them. Formal syntactic differences lead to create two new entries for this verb, in classes 35R and 35S. The entry in class 35S refers to 'the state of not being in good relations with someone' and it can be considered as an infrequent use, since only very few instances were found. The entry of class 35R is part of the semantic group of class 32R and the main difference is that it describes the process of communication in/using a given language and the verb is followed by the preposition *em*.
  The ViPEr database was reviewed and the semantic group that refers to the communication process is currently composed by three different constructions: 41, 32S and 35S. This set of word senses was differentiated from the other semantic group using Portuguese-English paired sentences and the tags *<falar, mention>* and *<falar, discuss>*. As the use from class 35S is very infrequent, only the other two classes were annotated using scripts

with two, very simple, heuristics. When the verb was followed by *de*, *do*, *da*, *em*, *no*, *na* or *sobre* in the following two words, then it was classified as an instance of class 41. If the verb was followed by the preposition *com* in the following two words, then it was classified as an instance of class 42S construction.
Semantically, the uses from classes 32R and 35R are closely related and there was no translation tag that could identify them correctly. However, the tags mentioned and used previously had a hight probability of not corresponding to this semantic group. The most frequent tag <*falar,speak*> was used, combined with two scripts to differentiate both constructions. If the verb was followed by *língua*, *línguas* or *linguagem* on the following three words after the verb and the first word after the verb was the preposition *em*, then it was classified as 35R. A list of frequent languages' names was added to the previous words to increase the number of extracted instances. When the previous conditions were verified without the preposition *em*, then the class 32R was attributed.

- *pensar/think:* the different constructions of the verb *pensar* are semantically very similar and they are mainly distinguished by syntactic features alone. For this reason and to prevent deviations on the study, this verb was not considered.
- *explicar/explain:* The verb *explicar* has 2 frequent ViPEr constructions. The first is in class 09I, and it is used to denote a communication process, like the verb *talk* or *speak*. The second one is in class 01T, and it expresses a causal relation between two events/states. To better understand the difference, consider Example 4.

[09I] *O Pedro explicou as regras do xadrez ao João* (Peter explained the rules of chess to João)
[01T] *Isso explica o comportamento do Pedro* (That explains Peter's behavior)
Example 4. Different constructions of the verb *explicar* 'explain' considered in ViPEr.

The use from class 09I was identified using the combination of Portuguese with Italian. There were four different candidate tags that could be mapped onto the 09I construction: <*explicar, chiarire*>, <*explicar, illustrare*>, <*explicar, spiegarci*> and <*explicar, spiegarlo*>. Considering the last two tuples, this is the same verb combined with a clitic. It results from the use of the spaCy tool in the lemmatization process (see Section 2.3) and it does not correspond to the usual concept of lemma. However, this distinction revealed itself to be useful. As in Portuguese, the Italian translation of the verb *explicar* has also two constructions for the same verb, but the fact of the verb appearing with clitics enabled us to correctly distinguish at least one construction of this verb. Regarding the use represented in class 01T, some instances were extracted using a script. The class 01T was attributed when the following conditions were verified: the verb had the most frequent tag <*explicar, spiegare*>, it was preceded by the word *que*, *isto*, *isso* or *aquilo* or the following two words were *por*, *pelo*, *pela* or *porque*.

The mapping process from the previous tags onto the corresponding ViPEr verb classes allowed to acquire knowledge to create training files for 8 more verbs: *contar* 'count', *dar* 'give', *deixar* 'leave', *ganhar* 'win', *lembrar* 'remember', *ouvir* 'listen', *parecer* 'seem' and *passar* 'pass'. These verbs are those that have had a high percentage of global wrongly identified instances and that had not yet been annotated for their senses in *corpus*. As these verbs are very frequent on the evaluation corpus, they also had a higher impact on the WSD global results.

5 EVALUATION

To evaluate the performance of the new training files produced for the Naive-Bayes algorithm, the PAROLE *corpus* [6] was used. This *corpus* contains texts from different sources, such as journalistic texts from newspapers and prose from novels. It contains around 38,928 verbs, from which 13,108 are ambiguous verbs considered for the evaluation. Each verb was manually annotated by linguists with its ViPEr class. Two evaluation metrics were used: the precision and the accuracy.

The *precision* measure indicates the fraction of correctly identified instances of a verb among all the instances of that verb identified by the system. Equation 1 introduces the corresponding formula.

$$Precision = \frac{Number\ of\ correctly\ identified\ instances}{Number\ of\ identified\ instances} \quad (1)$$

The *accuracy* measure indicates how many verbs were correctly classified, among all the verbs present in the evaluation *corpus*. Equation 2 indicates the formula for the accuracy metrics, with n_c being the number of correctly disambiguated instances and N the total number of ambiguous and processable verb instances in the evaluation *corpus* (13,108).

$$Accuracy = \frac{n_c}{N} \quad (2)$$

5.1 First 4 verb models evaluation

In this step of the evaluation, the automatically annotated files regarding the first 4 verbs were reviewed by a linguist. Figure 3 demonstrates how many instances of each verb class where correctly or wrongly classified, when compared against the reviewed files. Generally, the precision is above 60%, and there are constructions, such as class 41 of the verb *falar* 'speak', that have above 90% of precision. Also, the time that is required to manually review or annotate all sentences is reduced to about half. In these pre-annotated *corpora*, the sentences are sorted by ViPEr classes, meaning that it is much easier to verify and automatically assign a classification than to think about all possible constructions for a given verb instance, especially when shown among other, randomly presented, instances of that verb.

In the next step, the PAROLE *corpus* was used to evaluate the previous set of verbs. Also, after manual verification, the *corpus* does not have instances of the new constructions of classes 35S and 35R for the verb *falar*, so it was not possible to evaluate the scripts' precision for these classes. Table 3 refers to the evaluation before using the new training files and it gives, for each verb, a general view of the total number of instance, the number of instances that

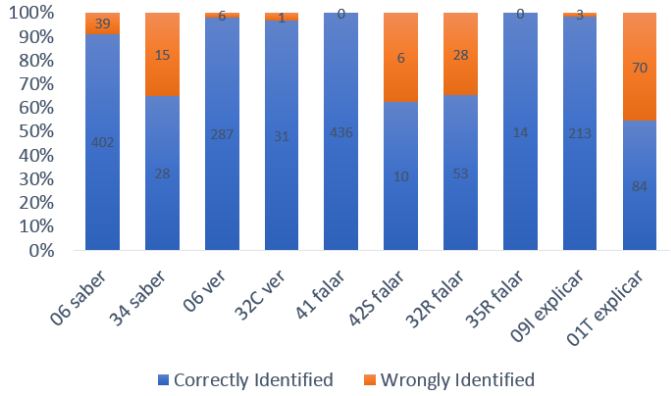

Figure 3: Annotations comparison against the reviewed files.

were correctly disambiguated and the corresponding precision. Regarding the verbs *saber*, *falar* and *explicar*, the precision was above 90%, meaning that it will be difficult to surpass this results with the new training files. The lowest precision value, 64.16%, was obtained with the verb *ver*.

Lemma	Instances	Correctly Disamb.	% Correctly Disamb.
saber	468	448	95.73
ver	438	281	64.16
falar	201	195	97.01
explicar	132	125	94.70
Total	1,239	1,049	-

Table 3: Rules + ML evaluation using former training files.

Lemma	Instances	Correctly Disamb.	Difference	% Correctly Disamb.
saber	468	446	-2	95.30
ver	438	217	-64	49.54
falar	201	195	0	97.01
explicar	132	121	-4	91.67
Total	1,239	979	-70	-

Table 4: Rules + ML evaluation using new training files not reviewed.

Table 4 contains the results for the evaluation, which used the new ML models that were trained over the new, not reviewed, annotated files. For the verbs for which a high value of precision had already been attained, the results were similar. Only the verb *ver* had an important reduction on the number of correctly identified instances, as more 64 verb instances have been wrongly identified, and a reduction of 15.07% in precision was observed, achieving only 49.54%.

Table 5 contains the same information as the Table 4, however the evaluation used the models trained over the files reviewed by the linguist. Considering the results, only the verb *explicar* improved the precision, exceeding the reference results by one instance. This proves that the effort necessary to review the sentences did not compensate, as the results were almost the same. This can be justified by the previously described Figure 3, as the deviation of the new files from the reviewed files does not have a significant effect on the Naive-Bayes models.

Lemma	Instances	Correctly Disamb.	Difference	% Correctly Disamb.
saber	468	446	-2	95.30
ver	438	217	-64	49.54
falar	201	195	0	97.01
explicar	132	126	+1	95.45
Total	1,239	984	-65	-

Table 5: Rules + ML evaluation using new training files reviewed.

Lemma	Instances	Correctly Disamb.	Difference	% Correctly Disamb.
saber	468	448	0	95.73
ver	438	271	-10	61.87
falar	201	195	0	97.01
explicar	132	126	+1	95.45
Total	1,239	1,040	-9	-

Table 6: Rules + ML evaluation using former and new training files reviewed.

Finally, Table 6 contains the results of combining old and new reviewed training files to train the models for the ML algorithm. In this case, the results for the verb *ver* had a significant improvement, achieving 61.87% of precision. However, it does not surpass the original training files, as there are more 10 instances wrongly identified. This proves that the method used for annotating the new training files produce similar results as manually annotating all sentences from scratch.

5.2 New verb models evaluation

The second part of the evaluation consisted in evaluating the performance of the 8 new Naive-Bayes models that were created using the new training files. First, the system performance with the new training files is compared with the results of evaluating the system with the rules combined with the Most Frequent Sense (MFS). Table 7 gives a general view of the results of evaluating the rule-based disambiguation combined with the MFS, for each of the annotated verbs.

Table 8 contains the results regarding the evaluation made using rules combined with ML disambiguation. The new Naive Bayes models were trained over files that contain sentences annotated

Lemma	Instances	Correctly Disamb.	% Correctly Disamb.
contar	127	69	54.33
dar	107	51	47.66
deixar	119	69	57.98
ganhar	58	10	17.24
lembrar	81	12	14.81
ouvir	133	99	77.44
parecer	47	13	27.66
passar	158	49	31.01
Total	830	372	-

Table 7: Rules + MFS evaluation.

Lemma	Instances	Correctly Disamb.	Difference	% Correctly Disamb.
contar	127	81	+12	63.78
dar	107	60	+9	56.07
deixar	119	87	+18	73.11
ganhar	58	47	+37	81.03
lembrar	81	60	+48	74.07
ouvir	133	106	+7	79.70
parecer	47	43	+30	91.49
passar	158	72	+23	45.57
Total	830	556	+184	-

Table 8: Rules + ML evaluation.

using the methodology mentioned before. This files have not been reviewed by a linguist, as the preliminary experiments did not showed significant improvements after the revision. To compare the precision of both models, Figure 4 presents the precision values of processing the evaluation corpus with the Rules combined with the MFS and the precision when processing the rules combined with ML-disambiguation.

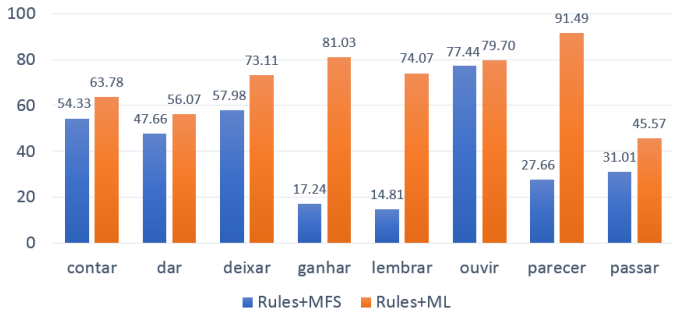

Figure 4: Comparison of precision of Rules+MFS against the Rules+ML.

Generally, the number of correctly disambiguated instances improved for all verbs and the best results were achieved with verbs that have had low precision values on the previous classifier. The highest improvements were on verb *lembrar* 'remember', that passed from 12 to 60 (+48) correctly disambiguated instances, and on verb *ganhar* 'gain', that passed from 10 to 47 (+37) correctly disambiguated instances. The poorer results were seen for verbs *ouvir* 'listen' and *dar* 'give'. For the verb *ouvir* 'listen', as it only contains two different and frequently occurring classes, this result is related with the classes distribution on the training file. Regarding the verb *dar* 'give', the result is mostly related to the high number of potential senses and their absence in the training file. From 7 frequently occurring ViPEr classes, only 3 of them were present and annotated on the training file.

Lemma	Instances	Correctly Disamb.	Difference	% Correctly Disamb.
contar	127	64	-5	50.39
dar	107	48	-3	44.86
deixar	119	77	+8	64.71
ganhar	58	10	+0	17.24
lembrar	81	44	+32	54.32
ouvir	133	107	+8	80.45
parecer	47	13	+0	27.66
passar	158	51	+2	32.28
Total	830	414	+42	-

Table 9: Results using only ML.

Another experiment consisted in removing the rules and evaluating the functionality using only the ML-disambiguation (Table 9). The produced results were much worse, as there are multiple verbs for which the number of correctly disambiguated instances was lower than the one produced using the old classifier, this is, the rules combined with the MFS. While the rules combined with ML correctly identified more 184 instances, the ML alone was only capable to correctly identify more 42 instances. Only the verb *ouvir* 'listen' produced slightly better results, surpassing the rules combined with ML by one instance. This experiment has proved that rules are important to the VSD task.

5.3 Comparison against baseline

The performance of the system was also compared against a *baseline*, a reference performance threshold. In this case, the MFS (Most Frequent Sense) classifier was used as a baseline, as it is the most basic approach on the WSD tasks. This classifier is based on simple counting. In the training step, the counts of occurrences of each sense for each lemma are produced. Then, when the verb model is classifying, the most frequent sense is assigned to every instance of that lemma. This model always classifies every instance of a given lemma with the same class (the most frequent sense). The counts were produced based on the PAROLE *corpus*, meaning that there may be a bias in this classifier. Figure 5 presents the precision of processing the evaluation *corpus* with the MFS and the precision when processing the rules combined with ML-disambiguation.

The precision values improved for all verbs, except the verb *ouvir*, though the difference in this case is minimal (0.75%). There are verbs, such as *parecer*, that passed from 72.34% to 91.49% (+19.15%). Regarding the verb *ouvir*, however, the similar values could be justified by the fact that this verb has only 2 senses, with one of

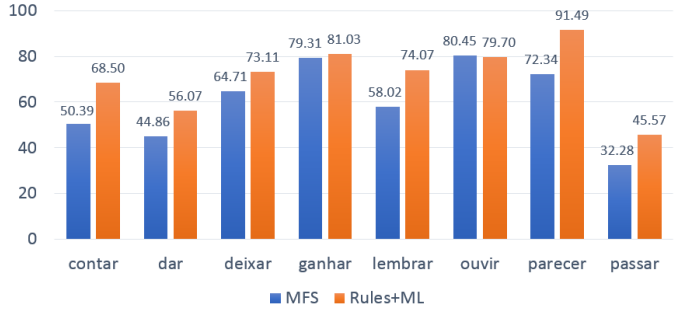

Figure 5: Comparison of precision of MFS against the Rules+ML.

them with a high number of instances. In this scenario, it is difficult to surpass the baseline.

5.4 Coverage

The verb sense disambiguation process requires dealing with a multiple classes' scenario, and depending on the input context, the expected output could be much different. In that way, Table 10 allows to understand the difference between the previous disambiguation method, which used only the rules combined with the MFS, and the new one that uses the ML disambiguation instead of MFS and was applied to the previously annotated verbs. The first two columns indicate for each verb, the number of different classes identified on the evaluation *corpus* with each of the classifiers. The last two columns gives the real number of different classes on the evaluation *corpus* and on the ViPEr database. The table shows that even the PAROLE *corpus*, which is composed of texts from different sources, often does not contain all classes that were defined in the ViPEr database.

Verb	Rules + MFS	Rules + ML	PAROLE	ViPEr
contar	2	5	7	8
dar	2	3	8	11
deixar	2	3	5	5
ganhar	1	2	4	5
lembrar	2	2	3	3
ouvir	1	2	2	2
parecer	1	2	2	3
passar	2	4	11	11
Total	13	23	42	48

Table 10: Number of different classes identified on the evaluation *corpus*.

Note that the values regarding the real number of different classes contain a high number of rarer construction, meaning that they were not considered on the training files and also they do not have a significant impact on the system performance. It is possible to conclude that the new classifier covers many more different classes. Generally, the precision not only improved for all verbs, as the models have a larger lexical coverage. Only the verb *lembrar* maintained the number of different classes that were previously identified.

5.5 Discussion

In the previous sections, different scenarios were used to evaluate the performance of the new Naive Bayes models trained over the new training files. It was possible to compare these models performance against the former classifier, as well as with the MFS. The produced results not only showed better precision values, but they also proved that the new models have a larger lexical coverage.

Classifier	Accuracy
Rules+MFS+ML	80.50
Rules+MFS+ML (with new models)	82.07

Table 11: STRING's performance before and after adding the new models.

To conclude the evaluation process, the system's general performance was evaluated. For that purpose, the accuracy of both classifiers was measured, and results are shown in Table 11. The accuracy of the system improved from 80.50% to 82.07%, an increase of 1.57%. These results allow us to conclude that the new models constitute a positive contribute for the STRING performance regarding the VSD task, as 1.57% correspond to more 206 sentences correctly disambiguated.

6 CONCLUSIONS

This paper proposed an approach to semi-automatically produce training *corpora* annotated with the verbs senses, to be used with (supervised) Machine Learning (ML) techniques in the WSD task in STRING.

First, the data was prepared for the Tagging Module. For that, some pairwise pre-processing steps were applied to align, at sentence level, each of two paragraph-aligned texts. The pair of generated aligned sentences was composed by a Pivot Language (PL) file and by a foreign language file, which was used for disambiguation. For this project, Portuguese was used as PL and 5 other foreign languages were used for the alignments: English, Spanish, French, Italian and German. These specific languages reduced the loss of information in the alignment process and maximized the amount of available data.

After the pre-processing stage, the 5 previously aligned bitexts were pooled together in different combinations, using the PL as a bridge. This process had a challenge associated, as the alignment process faces the particularity that there are sentences in one of the bitexts that cannot be mapped onto any sentence of the other bitext, and must therefore be removed. For that purpose, an algorithm was developed.

The second step of data preparation was the generation of the word mapping tables. The GIZA++ was used to create a mapping between the words of two (sentence-aligned) texts.

A new Tagging Module was implemented, which is responsible for tagging the sentences in the PL with multilingual tuples, and some experiments regarding the module functionality were

explored. To elicit insights and draw tentative conclusions about the variables and values to be used on the tagging process, the Inverse Phrase Translation Probability (IPTP) and Inverse Lexical Weighting (ILW) values were successively adjusted, starting with 0, and then increasing them until at least the prepositions were eliminated from the resulting tags. It was concluded that establishing a minimal value of 0.3 for IPTP and of 0.001 for ILW discards almost all the errors on the alignment process. Also, a lemmatization process was used to reduce the number of multiple verb forms belonging to the same lexical entry and to eliminate word classes that were not verbs.

As it could not be known beforehand which language combinations would more appropriate for disambiguating any given verb, all possible alignments were generated. The number of aligned sentences in the corpus is reduced by about 200 thousand sentences for each extra language aligned. It was concluded that using only one or at most two languages could produce the best results. If a given language is not able to distinguish the verb meanings by different translations, the probability of distinguishing them by combining more languages is significantly reduced.

In the next step, when the tagged data was generated, the tags were manually mapped onto ViPEr classes. The task of analyzing whether a given tag was able to disambiguate the existent ViPEr classes required semantic knowledge that, at this stage, could only be provided by a human.

When the first 4 training files were evaluated against the old files, the results were almost the same, except for the verb *ver* 'see'. Also, the new training files were reviewed by a linguist, and on the evaluation almost the same results were achieved. This proved that the effort necessary to review the automatically pre-annotated sentences did not compensate, as the produced results are very similar. Furthermore, the deviation of the new files from the reviewed files did not have a significant effect on the Naive-Bayes models. Nevertheless, the time that is required to manually review all pre-annotated sentences is reduced to about half, as the sentences are sorted by ViPEr classes of the target verbs.

The previously acquired knowledge was used to create training files for 8 more verbs. These verbs are those that have had a high percentage of global wrongly identified instances and that had not yet been annotated for their senses in *corpus*. As these verbs are very frequent on the evaluation corpus, they also had a higher impact on the WSD global results.

To conclude, the system performance regarding the correctness of the produced results was measured by the precision and accuracy evaluation metrics. The results of evaluating the system performance with the new training files was compared with the results of evaluating the system with the rules combined with the MFS. The number of correctly disambiguated instances improved for all verbs and the best results were achieved with verbs that had low precision values on the previous classifier. The accuracy of the system improved from 80.50% to 82.07%, an increase of 1.57%. When the new models were evaluated against the baseline, the MFS classifier alone, the precision values generally improved. Also, the new models have a broader lexical coverage. This is a good result, as it is preferable to have a system which is capable to adapt to different scenarios than to have a system that is optimal with a specific type of input.

REFERENCES

[1] Jorge Baptista. 2005. Construções simétricas: argumentos e complementos. *Estudos de Homenagem a Mário Vilela* (2005), 353–367.

[2] Jorge Baptista. 2012. ViPEr: A Lexicon-Grammar of European Portuguese Verbs. In *Proceedings of the 31st International Conference on Lexis and Grammar*. Nové Hrady, Czech Republic, 10–16.

[3] Jorge Baptista. 2013. ViPEr: uma base de dados de construções léxico-sintáticas de verbos do Português Europeu. In *XXVIII Encontro Nacional da Associação Portuguesa de Linguística*. 111–129.

[4] Jorge Baptista and Nuno Mamede. 2013. Reciprocal Echo Complements in Portuguese: Linguistic Description of in view of Rule-based Parsing. In *Proceedings of the 32nd International Conference on Lexis and Grammar*. Faro, Portugal, 33–40.

[5] Philipp Koehn. 2005. Europarl: A Parallel Corpus for Statistical Machine Translation. In *Proceedings of the Tenth Machine Translation Summit (MT Summit X)*. Phuket, Thailand.

[6] M. Nascimento, P. Marrafa, L. Pereira, R. Ribeiro, R. Veloso, and L. Wittmann. 1998. LE-PAROLE - Do corpus à modelização da informação lexical num sistema-multifunção. In *XIII Encontro Nacional da Associação Portuguesa de Linguística*. Lisboa: APL/Colibri, 115–134.

[7] Nuno Mamede, Jorge Baptista, Cláudio Diniz, and Vera Cabarrão. 2012. STRING: A Hybrid Statistical and Rule-Based Natural Language Processing Chain for Portuguese. In *Proceedings of the 12th International Conference on Computational Processing of the Portuguese Language - Demo sessions* `http://www.propor2012.org/demos/DemoSTRING.pdf` *(PROPOR 2012)*.

[8] Roberto Navigli. 2009. Word sense disambiguation: A survey. *ACM Computing Surveys (CSUR)* 41, 2 (2009), 1–69.

[9] Franz Josef Och and Hermann Ney. 2003. A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics* 29, 1 (2003), 19–51.

[10] Ricardo Pires. 2017. *Verb Sense Disambiguation in STRING*. Master's thesis. Instituto Superior Técnico, Universidade de Lisboa.

[11] Ahmad Shahid and Dimitar Kazakov. 2010. Retrieving Lexical Semantics from Multilingual Corpora. *Polibits* 41 (2010), 25–28.

[12] Gonçalo Suíssas. 2014. *Verb Sense Classification*. Master's thesis. Instituto Superior Técnico, Universidade de Lisboa.

[13] Tiago Travanca. 2013. *Verb Sense Disambiguation*. Master's thesis. Instituto Superior Técnico, Universidade de Lisboa.