Using unsupervised word sense disambiguation to guess verb subjects on untagged corpora

Paula Cristina Vaz and David Martins de Matos

Spoken Language Laboratory
INESC-ID-Lisboa
Lisboa, Portugal
paula.vaz,david.matos@l2f.inesc-id.pt

Abstract. This article explores the use of subject lists extracted from an annotated corpus to find subject-verb pairs in untagged corpora. Our goal is to identify verb syntactic functions (subjects and direct objects) to characterize verb arguments. Since identifying syntactic functions on corpora using parsers is time-consuming, it is desirable to automate the annotation process of the syntactic functions without parsing the corpus. We present a method that uses a small annotated corpus to cluster sentences with synonymous verbs. We observe that verbs in the same cluster have the same list of nouns as subject in the test corpus, even though the specific pair subject/verb does not appear in the annotated corpus. The result shows that annotating the subject/verb pair using the subject lists extracted from the clusters is quicker than syntactically parsing the corpus.

1 Introduction

Our goal is to find nouns that can be used as subjects or objects using a small annotated manually-corrected corpus and without having to parse large amounts of corpora. Syntactically parsing usually means to lemmatize, disambiguate, chunk or find the parse tree, and, finally, connect the subject, object, and prepositional object with the predicate of the phrase or sentence. After the parsing process is complete, the corpus is ready to be searched for (subject, predicate), (object, predicate), or (preposition, predicate) pairs. Performing all these tasks is time-consuming. It becomes desirable to find a method of extracting syntactic functions without parsing the corpus.

A syntactically annotated corpus for the Portuguese language, Bosque Sintáctico [1], is publicly available. To our knowledge, this corpus is the only one manually-corrected. This corpus is small (180k words) and the number of phrases available for each predicate is also small, e.g., 52 phrases with `viver` (to live) and only 5 with `habitar` (to inhabit). Moreover, some of the phrases use pronouns or proper nouns as subjects and do not contribute with useful information.

We observed that synonym verbs have the same nouns as possible subjects, objects, or prepositional objects (example 1).

Example 1. The predicate in sentence 1.1 has the subject `equipa` (team) and is a synonym of the predicate in sentence 1.2. The subject in sentence 1.2 is `fabricante` (manufacturer).

1.1 A equipa `mostra` progressos...
The team `shows` progress...

1.2 O primeiro fabricante de ratos para computador `apresentou` esta semana...
The first manufacturer of mouse for computer `presented` this week...

Since the two predicates are synonyms, it is possible to switch predicates between the two sentences as shown in 1.3 and 1.4

1.3 A equipa `apresenta` progressos...
The team `presents` progress...

1.4 O primeiro fabricante de ratos para computador `mostrou` esta semana...
The first manufacturer of mouse for computer `showed` this week...

Acquiring `equipa` (team) as a possible subject for the predicate `apresentar` (present) and `fabricante` (manufacturer) as subject of `mostrar` (show) verb.

We also observed that some predicates have nouns with a higher semantic probability of being subjects, objects, or prepositional objects then others (example 2).

Example 2. We can assume that the probability of having the noun `chocolate` as subject is higher for the predicate `melt` (2.1), then for the predicate `sing` (2.2).

2.1 Chocolate `melted` inside the oven.

2.2 Chocolate was `singing` on top of the stairs.

Based on these two properties, sentences with synonym predicates can be clustered, and we can generate lists of possible subjects for each predicate in the clusters. We present a method that uses a small corpus and unsupervised clustering to find subjects for synonym predicates.

2 Word Sense Disambiguation

The WSD problem is well-known and widely studied by the natural language processing (NLP) community. Ide and Véronis [2] point out that WSD is not a

useful task on its own, but rather an "enabling technology" for NLP [3]: typically, they are expected to be useful components of information retrieval, question answering, document classification, and machine translation applications, among others. Nevertheless, it is also agree that it is one of the most difficult problems to solve [2]. Its major difficulty lies in finding good sense inventories [4] and, when they can be found, sense classifications are hand-crafted by human linguists, lacking the objectivity desirable for automated processing: different linguists tend to make classifications based on personal knowledge and experience, thus imposing an idiosyncratic interpretation on word senses [5].

Various WSD methods exist. They can be classified, according to their needs concerning previous sense-tagged data and resources, as knowledge-based, supervised, and unsupervised [3].

2.1 Knowledge-based and supervised approaches

Knowledge-based methods depend on some sort of knowledge source (e.g. the Lesk algorithm [6] and semantic similarity [7]). Knowledge sources used by these approaches can be dictionaries, rules, or heuristics. Supervised corpus-based methods typically use supervised machine learning algorithms trained on manually tagged corpora or bootstrapping algorithms from seed data (semi-supervised).

Although supervised corpus-based methods have better performance than knowledge-based methods, the latter have larger coverage. Knowledge-based methods have the disadvantage of requiring a knowledge source for providing the word senses, but sense inventories and rules databases are not always available. Any approach to WSD that is dependent on sense inventories is permanently locked into a fixed view of word senses that will not evolve or adapt with time and will have a high degree of language dependence. Supervised corpus-based algorithms have the disadvantage of requiring tagged corpora, even in small amounts.

2.2 Unsupervised word sense disambiguation

Unsupervised approaches are useful when sense tagged corpora are not available. Also, more generic systems, i.e., that are not committed with specific sense domains (e.g. medical, biotechnological, etc.), benefit from unsupervised approaches since they cannot be dependent on a particular sense inventory.

Another advantage of using unsupervised approaches is the permanent availability of corpora. If we consider the World Wide Web, texts about almost any subject are available virtually in unlimited amounts.

In spite of the above mentioned advantages, unsupervised WSD methods are, however, more difficult to implement, because learning algorithms do not have reference corpora to compare results. Moreover, establishing the number of senses present for each word being disambiguated can be a problem, and performance is almost always inferior to that of supervised or knowledge-based systems.

Typically, unsupervised methods use statistical algorithms on raw data and may be knowledge-lean [8]. Unsupervised WSD systems that are not knowledge-lean use some knowledge source to estimate the possible number of senses for each word being disambiguated. One known approach is to use bilingual aligned corpora to distinguish senses that have different translations between two languages. Another approach is to use thesaurus entries as tags for sense clusters.

Nevertheless, completely knowledge-lean WSD is possible. This type of methods uses similarity measures to cluster together sense-related words or sentences (contexts) and similarity measures are calculated based on word usage.

Pedersen [8] arranges knowledge-lean methods in two classes: type-based methods and token-based methods. Type-based methods cluster together words that are related by use in similar contexts. These methods create a representation of different words in a corpus that attempts to capture their contextual similarity. Representations are usually based on counts of co-occurrences or measures of association between words.

Token-based methods cluster together contexts with a target word based on the similarity of the contexts. The resulting clusters are composed by contexts that use the target word in the same sense. Token-based methods are usually based on first- and second-order features. First-order features occur directly in a context, but second-order occur with a first-order feature, bur may not occur in the context.

Type- and token-based methods can be combined. A type-based method is used to cluster words in sets. These sets of related words only depend on the nature of corpora and are a source of features. A token-based method is then used to cluster contexts based on the sets of related words. Context clusters will be labeled with the words used as features [9, 10].

2.3 SenseClusters

SenseClusters is a freely available open-source system that works under the hypothesis that words that occur in similar contexts will have similar meanings. SenseClusters represents the instances to be clustered using second order co-occurrence vectors. These are constructed by first identifying word co-occurrences, and then replacing each word in an instance to be clustered with its co-occurrence vector.

We configured SenseClusters to make token-based clustering of contexts using bigrams as features to define the clusters. Bigrams are calculated with token word. The number of clusters is discovered automatically using PK2 statistics [11] and are constructed using the repeated bisections clustering method.

3 An automated method to guess verb subjects on untagged corpora

Our method uses a small tagged corpus (180k words) and a large untagged newspaper corpus (6M words). The system is subdivided in two subsystems: the subject finder (figure 1) and the subject tagger.

The subject finder is designed to preform three main steps: the first uses the lemmatized annotated corpus to create feature files and sets of sentences; the second uses SenseClusters with the feature files to cluster sense-related sentences; finally, the last step expands subject sets of all the verbs in the same cluster creating the acquired subjects file. The subject tagger system uses the acquired subjects file to search for possible subjects in the untagged corpus and tags them.

3.1 Corpus

Floresta Sintáctica (FS) [1] is a Portuguese language treebank created from CETEMPúblico [12], a collection of articles from the Portuguese daily Público. FS has 41,406 syntactic trees and about 1 million words, automatically annotated using PALAVRAS [13]. A subset of FS, the first 184,773 words, were manually corrected. This subset is called Bosque Sintáctico (BS) [1]. We use BS as our source of pre-annotated subject-verb pairs. In addition, we used Palavroso [14] and MARv [15] to lemmatize the annotated corpus.

3.2 Extracting and expanding subjects

The subject finder is composed by two main modules and SenseClusters, as shown in figure 1. The first module, the feature and sentence extractor (FSE), reads the tagged corpus and extracts the sentences with the referenced verbs and its subjects; SenseClusters then clusters the sentences; and, finally, the subject expander module "guesses" verb subjects in the same cluster (i.e. with the same sense).

Feature and Sentence Extractor This module uses sets of sense-related verbs (synonym sets) to select sentences from the corpus. The sets include verbs that appear in the same entry in a Portuguese synonyms dictionary.

Verb synonym sets do not include more than five elements, as shown in example 3. Our aim is to extend each set to all the verb synonyms of the corresponding dictionary entry.

Example 3. The synonym set for verb `viver` includes `viver` (to live), `morar` (to dwell), `habitar` (to inhabit), and `residir` (to occupy a residence).

The FSE searches the annotated corpus for sentences containing verbs in each the verb synonym set and generates two files. The first is the "dummy-file" and contains all the lemmatized sentences selected from the corpus. FSE replaces verbs in these sentences, belonging to the verb synonym set, with a tag word (`dummy`) as shown in example 4.

The second file is the "feature-file" and contains the nouns of the extracted sentences tagged as verb subjects (example 5).

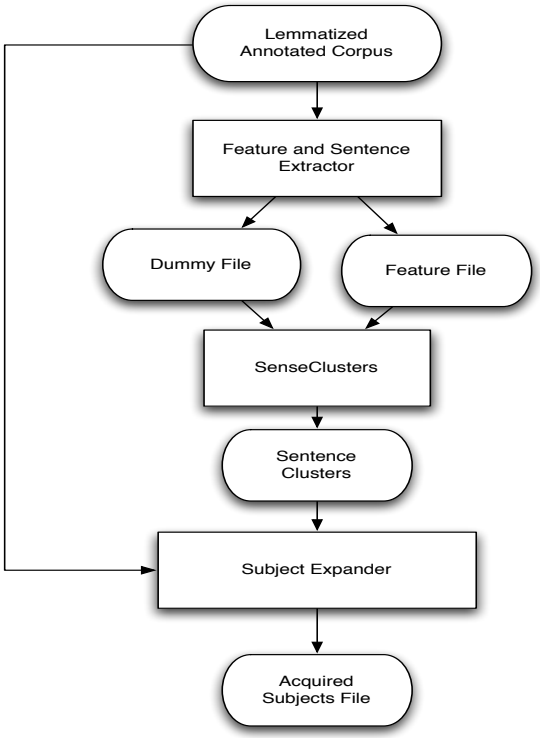

Fig. 1. Architecture of the subject finder system

Example 4. Verb `viver` on the sentence 4.1 is replaced by the tag `<head>dummy</head>` as follows,

4.1 o animal doméstico que `viver` em o campo ter um existência mais feliz
the house animal that `live` in the countryside have existence more happy

4.2 o animal doméstico que `<head>dummy</head>` em o campo ter um existência mais feliz
the house animal that `<head>dummy</head>` in the countryside have existence more happy

Example 5. For the verb `viver` (to live) the system found the sentences with subjects `banco` (bank) and `Manuel` (proper name) among others.

SenseClustering verb senses SenseClusters clustered sentences according to each verb's possible senses. We configured SenseClusters to use the PK2 cluster stopping measure [11] and co-occurrences.

The remaining parameters were set according to Kulkarni for a similar type of problem [11] (I2 for criterion function; similarity measures; repeated bisection as clustering method; likelihood for statistical test). The reason for using co-occurrences, is that, although subjects typically appear before the verb, they can also be placed after the verb.

We generate two sets of clusters: one set with the feature file as the SenseClusters training corpus and the other without training.

Expanding subjects The subject expander uses the dummy-clusters file and the original sentences to replace the dummy tag with the original verb. Each cluster has sentences with semantically equivalent verbs. In theory, we could replace predicates of each sentence by any of the verbs allocated to the cluster without changing its meaning, as shown in example 6.

Based on the heuristic that synonym verbs have the same subjects, the subject expander merges subjects and verbs included in the same cluster. Thus, subjects of a verb are expanded to others in the same cluster, i.e., with the same sense, and written to the acquired subjects file.

Example 6. In the sentence 6.1, verb `morar` may be replaced with the synonym `viver` (sentence 6.2) without changing the meaning of the sentence.

6.1 A Maria `mora` com o marido
Mary `dwells` with her husband
6.2 A Maria `vive` com o marido
Mary `lives` with her husband

3.3 Subject Tagger

The subject tagger (figure 2) reads the acquired subjects file and the untagged corpus that was previous lemmatized. Then, it searches and tags the sentences that contain the pair subject-verb within a given window, as shown in example 7.

Example 7. Results for the acquired subject `emigrante` (emigrant), the verb `viver`, and a window of 2 words are shown in sentences 7.1, 7.2, and 7.3. With the window size set to 2 words, the tagger searches for all the sentences with zero, one, or two words between the subject and the verb's lemma.

7.1 Window of 0 words
o emigrante viver o campeonato
(the emigrant live the championship)
o `<propsubj>`emigrante`</propsubj>` `<verb>`viver`</verb>` o campeonato

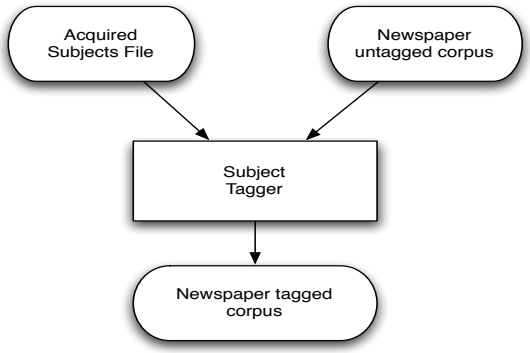

Fig. 2. Subject Tagger

7.2 Window of 1 word
o emigrante a viver no luxemburgo
(the emigrant that live in luxemburgo)
o <propsubj>emigrante</propsubj> a <verb>viver</verb> no luxemburgo

7.3 Window of 3 words
o emigrante cubano que viver em país
(the emigrant cuban that live in country)
o <propsubj>emigrante</propsubj> cubano que <verb>viver</verb> em país

4 Evaluation

We randomly chose three sets of predicate synonyms to evaluate the system: `viver` (to live), `andar` (to walk), and `dizer` (to say) synonym sets. The system could generate 293 pairs of acquired subjects for this synonym sets. Table 1 shows the number of subjects extracted from the annotated corpus in the "Original" column, and the number of acquired subjects (in the "Acquired" column) retrieved by the system.

Using the acquired subject list, the system could annotate 1484 subject-predicate pairs in the untagged corpus and 76.23% of this sentences were correctly annotated. Moreover, in the list of acquired subject-verb pairs we could find, at least, one sentence that used the subject-verb pair for 72.01% of those pairs.

Table 1. Number of predicate subjects.

Predicate	Original	Acquired
alegar	4	51
dizer	56	15
exprimir	2	53
falar	9	38
andar	6	42
circular	3	46
passar	42	9
passear	1	23
povoar	1	6
residir	3	6
viver	6	4
Total	133	293

5 Conclusions

We presented a method that uses an annotated corpus and unsupervised clustering to find verb subject pairs. Sentences with synonymous predicates were extracted from the tagged corpus. These sentences were clustered so that each cluster was composed of predicates with the same sense. We observed that verbs in the same cluster have the same list of nouns as subject, even though the specific pair subject-verb does not appear in the annotated corpus. Using this heuristic we expanded the combination of subject-verb pairs.

Although there is room for improvement, results show that this method can be used to annotate subject-verb pairs using the subject lists extracted from the clusters and without parsing the whole corpus.

Acknowledgments

We thank Ted Pedersen for his on line help with the use of SenseClusters.

References

1. Afonso, S., Bick, E., Haber, R., Santos, D.: Floresta sint(c)tica: a treebank for Portuguese. In Rodrigues, M.G., Araujo, C.P.S., eds.: Proceedings of LREC 2002, the Third International Conference on Language Resources and Evaluation, Paris, ELRA (2002) 1698–1703
2. Ide, N., Véronis, J.: Word sense disambiguation: The state of the art. Computational Linguistics **24** (1998) 1–40
3. Agirre, E., Edmonds, P.: Word Sense Disambiguation: Algorithms and Applications (Text, Speech and Language Technology). Springer-Verlag New York, Inc., Secaucus, NJ, USA (2006)

4. TufiŞ, D., Ion, R., Ide, N.: Fine-grained word sense disambiguation based on parallel corpora, word alignment, word clustering and aligned wordnets. In: COLING '04: Proceedings of the 20th international conference on Computational Linguistics, Morristown, NJ, USA, Association for Computational Linguistics (2004) 1312
5. Pedersen, T., Kulkarni, A.: Unsupervised discrimination of person names in web contexts. In: CICLing. (2007) 299–310
6. Lesk, M.: Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In: SIGDOC '86: Proceedings of the 5th annual international conference on Systems documentation, New York, NY, USA, ACM Press (1986) 24–26
7. Rada, R., Mili, H., Bicknell, E., Blettner, M.: Development and application of a metric on semantic nets. IEEE Transactions on Systems, Man and Cybernetics **19** (1989) 17–30
8. Pedersen, T.: Unsupervised corpus-based methods for wsd. In: Word Sense Disambiguation: Algorithms and Applications. Volume 33 of Text, Speech and Language Technology. Springer, Dordrecht, The Netherlands (2006) 133–166
9. Schütze, H.: Dimensions of meaning. In: Supercomputing '92: Proceedings of the 1992 ACM/IEEE conference on Supercomputing, Los Alamitos, CA, USA, IEEE Computer Society Press (1992) 787–796
10. Schütze, H.: Automatic word sense discrimination. Comput. Linguist. **24** (1998) 97–123
11. Kulkarni, A.: Unsupervised Context Discrimination and Automatic Cluster Stoppin. Master's thesis, University of Minesota, USA (2006)
12. Santos, D., Rocha, P.: Evaluating CETEMPublico, a free resource for portuguese. In: Meeting of the Association for Computational Linguistics. (2001) 442–449
13. Bick, E.: The Parsing System "Palavras": Automatic Grammatical Analysis of Portuguese in a Constraint Grammar Framework. PhD thesis, Aarhus University (2000)
14. Medeiros, J.C.: Processamento morfológico e correco ortográfica do português. Master's thesis, Instituto Superior Técnico – Universidade Técnica de Lisboa, Portugal (1995)
15. Ribeiro, R., Oliveira, L., , Trancoso, I.: Using morphossyntactic information in tts systems: comparing strategies for european portuguese. In: Computational Processing of the Portuguese Language: 6 *th* International Workshop, PROPOR 2003, Faro, Portugal, June 26-27, 2003. Proceedings. Volume 2721 of Lecture Notes in Computer Science., Springer (2003)