Evaluating Knowledge-poor and Knowledge-rich Features in Automatic Classification: A Case Study in WSD

Marcos Zampieri
University of Cologne
Albertus Magnus Platz 1, 50931
Cologne, Germany
Email: marcos.zampieri@uni-koeln.de

***Abstract*—Word Sense Disambiguation (WSD) is a fundamental task in many Computational Linguistics applications. It consists of automatically identifying the sense of ambiguous words in context using computational methods. This work evaluates the automatic disambiguation performance of five machine learning classifiers: Naive Bayes, Support Vector Machines, Decision Trees, KStar and Maximum Entropy. For the classification we compare the performance of these algorithms using knowledge-rich and knowledge-poor features applied to Portuguese data.**

I. Introduction

Many words have more than one meaning in natural language and the proper meaning is determined by the word's context. For example, the English word *appendix* can be defined in common use dictionaries as:

1) A separate part at the end of a book or magazine which gives additional information to readers.
2) A small tube-shape part which is joined to the intestines.

Any native or competent speaker of English will have no difficulty in understanding the correct sense of this word in contexts such as those presented in examples *a* and *b*:

a Please see the *appendix* for further details regarding the data used in the experiments.
b My daughter Sheila had a surgery to remove her *appendix* last summer.

However, when computational applications have to process these examples to accomplish tasks such as machine translation (MT) and speech recognition, this distinction is not always trivial. Statistical or rule-based methods are often used to help applications in distinguishing senses of words aimed at producing better results in language processing.

II. Background Information

Ambiguity has always been one of the major challenges for the computational processing of language. It is manifested in three levels of human language and each of them is addressed by a subtopic in computational linguistics.

At the morphosyntactic level, ambiguous words sometimes have different grammatical categories (e.g. the verb *to book* and the noun *book*). The process by which disambiguation occurs and correct grammatical categories are attributed is known as part of speech (POS) tagging [1].

Ambiguity can also be a purely syntactic phenomenon, related solely to the structure of the sentence. A famous example is the sentence: *The boy saw the man with the telescope*. This sentence allows at least two interpretations:

1) the boy saw the man using the telescope.
2) the boy saw the man who was carrying a telescope.

Cases like this are the object of the study of syntactic parsing [2]. Syntactic parsers identify sentence's constituents and the relations among them, using grammatical formalisms.

The third level of ambiguity, which is explored in this paper, is at the lexical-semantic level. At this level, words are ambiguous within the same part of speech, holding a homonymic or polysemic relation. The noun *appendix* is a clear example of this problem which is studied by a well-established field of research in computational linguistics called Word Sense Disambiguation (WSD) [3] and [4]. .

A. Approaches

The first studies on WSD were carried out in the 1950's as part of MT systems [5]. To address ambiguity at the lexical-semantic level and improve the output of translation software, early MT systems relied on a rule-based analysis module to assert senses of ambiguous words.

The disambiguation methods helped to improve performance in automatic translation and were later studied as an independent field of research that could be integrated in different language processing applications. Some applications of WSD include information retrieval (IR) [6] whereby words are disambiguated before being used in a search engine and speech processing systems [7] which aim to disambiguate homophonic and homographic words.

Lesk [8] proposed a method that used dictionary definitions, this was among the first approaches to WSD as an autonomous task. The method operated on the assumption that neighbouring words in a sentence would tend to share the same common topic or belong to related topics. Given an ambiguous word, the algorithm compares its dictionary definition with the definition of its neighbouring words in the

same sentence within a given interval. The resulting assigned sense is that definition having the highest number of words in common with the definitions of the neighbouring words. A later adaptation of the Lesk algorithm (with improved results) replaced dictionaries with Wordnet definitions [9]. Wordnet is a lexical database rich in semantic relations, and the approach was proposed by Banerjee and Pedersen [10].

Hirst [11] aimed to provide an abstract semantic representation of the entire input text, making it possible to distinguish senses of ambiguous words. Even though lexical ambiguity could be resolved by semantic representation, further studies have shown that this kind of approach is too ambitious and WSD should be modeled as a simpler task.

State-of-the-art methods in WSD do not rely on dictionaries for disambiguation. Following the work of Ng and Li [12], researchers started to use corpora as the main source of knowledge for disambiguation. Ng and Li's pioneering approach was described as an exemplar-based approach in which a word is assigned the sense of the most similar example already seen in the training stage. Their system is called LEXAS, a supervised learning approach which requires disambiguated text to be used as training data.

B. Machine Learning in WSD

Knowledge sources moved from dictionary definitions to corpora, and similarly, algorithms used for WSD have evolved. The first approaches were rule-based, whereas state-of-the-art research uses mostly statistical and machine learning techniques. The use of machine learning in WSD is often modeled as a classification problem and the features to be used vary according the proposed approach, common types include: morphosyntactic tags, neighbouring words in a window n, semantic tags and n-grams [13].

At this stage, one important distinction can be made between supervised and unsupervised approaches. Supervised approaches, as in this study, have a finite and predefined set of labels which correspond to the possible outcomes of the classification. A set of examples extracted from corpora and manually disambiguated is used to provide the system instances of training and as a gold standard. In unsupervised approaches disambiguated training examples are not provided and clustering techniques are used to group instances that belong to the same sense of the target word [14]. More recent approaches to unsupervised learning include [15] and more recently [16].

C. Scope of This Work

This paper compares the performance of five machine learning algorithms (Naive Bayes, Support Vector Machines, Decision Trees, KStar and Maximum Entropy) in disambiguating a set of Portuguese nouns using both knowledge-poor (simplistic) and knowledge-rich features. By simplistic, we mean that no additional information such as POS-tags or syntactic information was used for disambiguation. The classifiers used solely information available directly from the corpora. Preliminary work on the use of these features for Portuguese was carried out by Zampieri [17] and here we extend this method to different classifiers as section III-B describes.

The knowledge-rich features were obtained by enriching the corpus with morphosyntactic information using a POS Tagger [23]. This information as well as other lexico-syntactical features, was taken into account for disambiguation. We then compared the performance of the five classifiers using these two groups of features and evaluated to what extent POS and syntactic information helped the performance of the classifiers. The knowledge-rich features are better explained in section III-C.

We believe that the work presented here is an original contribution to the NLP research community for two main reasons. Firstly, because there are very few studies publish on WSD using Portuguese data. The most notable example the work of Specia [19], on applying WSD methods to Portuguese and English to increase performance in Machine Translation. The vast majority of studies on automatic disambiguation are applied to English data. Secondly, the comparison between the features proposed here in supervised classification provides a new outcome for Portuguese and all languages other than English.

III. Methods

The experiments started with the collection of a set of ten ambiguous nouns from a Portuguese vocabulary and establishing a catalogue of senses for each. As a starting point we used the Portuguese Academic Wordlist (P-AWL) [20] which is the Portuguese equivalent to the Academic Wordlist (AWL) [21] for English. These wordlists are widely used in experiments containing vocabulary and lexical entries in computational and applied linguistics.

The catalogue of senses was compiled based on common use dictionaries. We established two major senses according to corpus frequency and a third sense comprising all other occurrences of that word. We therefore modeled the classification scheme with three classes for each word (S1, S2 and S3) and the classifiers had to choose the correct class of a word in context.

The ten Portuguese words used for these experiments in alphabetic order were: *arquivo, crédito, cultura, essência, etiqueta, foco, garantia, geração, imagem* and *volume.*

A. Implementation

The sentences used to form our dataset were compiled based on a 2004 collection of the Brazilian newspaper *Folha de São Paulo*. In this corpus, texts are indentified by the name of the newspaper section that they belong to: Economy, Politics and Sports. This information was used for disambiguation using knowledge-poor features. We describe them in the next section.

B. Knowledge-poor Features

The idea of using knowledge-poor features was inspired by the work of Koeling et al. [22] which used the domain of the

text as the main resource for disambiguating English words. They claimed that for some domains, this information is enough for a classifier to assert the correct class of ambiguous words with a high success rate (above 80%). These features were later applied to Portuguese by [17] with satisfactory results.

The knowledge-poor features proposed here are divided in three groups:

1) Text Domain: We use the text domain as feature for disambiguation. As the corpus we used is already tagged with as meta-information, it was simple to extract. For corpora which do not contain this information, text classification techniques could be applied to make the use of this feature possible.

2) Neighbouring Words: Given an ambiguous word, the program fixed this word as an index and looked at a certain window to the left and to the right and used them as features. For this work we used the words that appear in a range of three. This feature can be a good source of information when applied to processed data where stop words have been removed.

3) Key Words: The key words features were extracted after frequency analysis of the data. The fifteen words that co-occur most often in sentences with the ambiguous word were considered. A Boolean value (true or false) was attributed to the presence or absence of each given word in the sentence.

C. Knowledge-rich Features

It is important to point out that in computational linguistics, knowledge-rich is often used to refer to features that add additional information to raw data. In WSD, this term has been used to describe featuresextracted from more sophisticated knowledge sources than those described here (POS Tags) such as Wordned synsets. We opted to use this term to contrast it to the simplistic or knowledge-poor features presented previously (those that do not use any external source of information for disambiguation).

There were two groups of knowledge-rich features used:

1) Lemmatized Word Bi-grams: Word n-grams are applied in several NLP applications to reveal co-occurrence patterns. The program counts the number of times words co-occur together in a corpus and uses this information as a feature for disambiguation. Here we use this the lemmatized version of each word. These patterns can indicate collocations and other syntactic characteristics of the data (e.g. fixed expressions or collocations that might accompany the ambiguous word).

2) POS Bi-grams: We annotated the corpus with POS information to be able to use this feature. The POS-Tagger used to annotate the corpus was the TreeTagger trained for Portuguese [23]. The POS corpus was then arranged in form of bi-grams which aim to provide information of grammatical patterns in the data. (e.g. the bi-gram "I am" would be represented as "P V" meaning Pronoun + Verb).

A snapshot of the tagset used in the annotation is presented in table number I.

Category	POS	Example
Adjective	ADJ	bonita
Adverb	ADV	muito
Determinant	DET	os
Cardinal	CARD	primeiro
Noun	NOM	mesa
Pronoun	P	eles
Preposition	PREP	de
Verb	V	fazer
Interjection	I	Oh!
Commas	VIRG	,
Punctuation	SENT	.

TABLE I
TREETAGGER TAGSET

D. Algorithms

The 5 algorithms compared in this study (Naive Bayes, Decision Trees, Maximum Entropy, Support Vector Machines and KStar) are widely used for WSD and differ substantially in their ways of performing classification. Here we used standard distributions of these five algorithms and no parameter was changed. Three of these classifiers are available in the Natural Language Toolkit (NLTK) [24], namely Naive Bayes, Decision Trees and Maximum Entropy and two of them available in WEKA Machine Learning Workbench [25]: Support Vector Machines and KStar.

An overview of each classifier based on what is described by [25] is presented as follows:

1) Naive Bayes: Based on Bayes Theory and Probability.

$$P(A|B) = \frac{P(A|B)P(A)}{P(B)} \quad (1)$$

Naive Bayes classifiers work under the assumption that the presence or absence of a particular feature of a class is not related to the presence or absence of any other feature. This independence assumption makes Naive Bayes classifiers particularly useful for supervised learning and makes the learning process faster than other learning algorithms.

2) Decision Trees: One of the most commonly used classifiers particularly for data mining. Decision Trees classifiers use trees to represent their models and arrange data into leafs and nodes each of which is assigned a given probability. It is widely, used especially due to the ease of understanding and interpretation of its tree structure representation.

3) Maximum Entropy: It is used as an alternative to Naive Bayes classifiers, since MaxEnt classifiers assume statistical dependence of its features and each of these estimations is calculated based on entropy model estimation. Due to this dependence assumption, Maximum Entropy classifiers tend to run significantly slower than Naive Bayes classifiers for the same set of data.

4) Support Vector Machines (SVM): SVMs are non-probabilistic binary classifiers. Given a set of instances, each of them belonging to one of two categories, SVM classifiers builds models that assign new examples to each class. An SVM model can be represented and understood as points in space. These points are mapped and the points belonging to

the two categories are usually as wide as possible to determine classification.

5) KStar: KStar or K* is an instance-based learner that uses entropic distance measure.

$$K^*(b|a) = -log_2 P^*(b|a) \quad (2)$$

It was developed prior to the WEKA Package [25] at the University of Waikato, New Zealand. [26] explain in detail how KStar performs classification and describe it as a lazy learner. For the simplistic features proposed in this work, a lazy learner can perform well and this will be discussed in the results section.

IV. Results

We report results in terms of precision, recall, f-measure and accuracy. For these experiments we used an average of 300 instances per word which were divided in partitions to be evaluated in a 3-fold cross validation scheme.

The metrics used are presented next are based on the results of a confusion matrix. The confusion matrix classifies results in 4 possible outcomes: tp, tn, fp and fn. Or *true positives*, *true negatives*, *false positives* and *false negatives*.

$$Precision = \frac{tp}{tp + fp} \quad (3)$$

$$Recall = \frac{tp}{tp + fn} \quad (4)$$

F-measure and accuracy provide a unified metric of success. F-measure takes precision and recall into account and it can be customized to emphasize one or another. In the formula used here, precision and recall have equal weights.

$$F - Measure = \frac{2PR}{P + R} \quad (5)$$

The fourth metric is accuracy (equation 6):

$$Accuracy = \frac{tp + tn}{tp + tn + fp + fn} \quad (6)$$

Accuracy takes into account the number of instances correctly classified ($tp + tn$) divided by all instances classified.

A. Knowledge-poor Results

The results using simplistic features are presented next and ordered from the best to the worst in terms of accuracy.

Classifier	Accuracy	Precision	Recall	F-Measure
Maximum Entropy	0.86	0.87	0.70	0.77
Naive Bayes	0.82	0.81	0.77	0.79
KStar	0.79	0.76	0.71	0.73
SVM	0.75	0.75	0.72	0.73
Decision Trees	0.70	0.75	0.69	0.72

TABLE II
KNOWLEDGE-POOR FEATURES

Best results reached 0.86 accuracy for Maximum Entropy and the worse 0.70 accuracy using decision trees. Naive Bayes had a slightly better performance in f-measure than Maximum Entropy, but still a lower accuracy and precision. Surprisingly, Support Vector Machines which are widely used in WSD research, for this group of features, did not score well.

B. Knowledge-rich Results

To compare the extend to which additional information improve performance in automatic classification, we performed disambiguation using knowledge-rich features. Classifiers are order according to their accuracy values.

Classifier	Accuracy	Precision	Recall	F-Measure
Maximum Entropy	0.81	0.80	0.70	0.75
SVM	0.78	0.77	0.75	0.76
Naive Bayes	0.76	0.74	0.74	0.74
KStar	0.72	0.71	0.72	0.71
Decision Trees	0.70	0.70	0.69	0.70

TABLE III
KNOWLEDGE-RICH FEATURES

The knowledge-rich features scored on average less than the classification with knowledge-poor features. The only algorithm that seems to benefit from this kind of feature is SVM, which was the second best for this group of feature and only the fourth using knowledge-poor. The lazy learner KStar had a substantially lower performance than on the first run and its accuracy results dropped in 7 percentage points from 0.79 to 0.72.

V. Conclusion

Results presented in this paper are an important perspective in supervised WSD research not only applied to Portuguese. The main conclusions of these experiments are:

- The use of features extracted exclusively from corpora with satisfactory results is a particularly interesting outcome for resource-poor languages: languages that do not possess the same amount of language engineering resources as POS Taggers and Parsers. There was no improvement of performance when using knowledge-rich features, suggesting that for this task, the information present on corpora might be enough for disambiguation.
- Using the text domain information proved to be an important informative feature in this study. This corroborates to the conclusions of Koeling et al. for English [22].
- Support Vector Machines seem to benefit from knowledge-rich features. SVMs need in general a more information to achieve good results, which is not possible by using solely simplistic features.

Even though many recent studies apply unsupervised classification to WSD, we believe that his does not invalidate the use of supervised approaches to this task. Both supervised and unsupervised methods have advantages and disadvantages: on one hand, it is possible to apply simple supervised methods to disambiguate a small pre-defined set of words as in the case of this paper. On the other hand, for more robust applications, unsupervised methods seem to be more suitable as they can deal with a bigger portion of the lexicon. In our opinion, both

approaches should be studied and improved depending on the needs of each software.

ACKNOWLEDGMENT

Early stages of this work were funded by a grant offered by the European Union Education and Training Commission, EMMC 2008-0083. The author thanks Constantin Orasan, Jorge Baptista and Rob Koeling for insightful comments on early stages of this paper and *Folha de São Paulo* for the texts used to form our corpus of study. Many thanks to the anonymous reviewers for their comments and suggestions.

REFERENCES

[1] E. Brill "Transformation-based error-driven learning and natural language processing: A case study in part-of-speech tagging" in *Computational Linguistics* 21, 1995, pp. 543 565.

[2] J. Carrol "Parsing" in R. Mitkov *Oxford Handbook of Computational Linguistics*, Oxford University Press, 2003, p. 233 248.

[3] E. Agirre and P. Edmonds *Word Sense Disambiguation: Algorithms and Applications*. Springer, 2006.

[4] M. Stevenson *Word Sense Disambiguation: The Case for Combination of Knowledge Sources*. CSLI Publications, 2003.

[5] M. Stevenson and Y. Wilks "Word Sense Disambiguation" in R. Mitkov *Oxford Handbook of Computational Linguistics*, Oxford University Press, 2003, pp. 249-265.

[6] A. Kulkarni and M. Heilman and M. Eskenazi and J. Callan "Word Sense Disambiguation for Vocabulary Learning" *Ninth International Conference on Intelligent Tutoring Systems*, 2008.

[7] D. Yarowsky Homograph disambiguation in text-to-speech synthesis. In J. van Santen, R. Sproat, J. Olive, and J. Hirschberg *Progress in Speech Synthesis*, Springer-Verlag, New York, 1997, pp.157-172.

[8] M. Lesk "Automatic Sense Disambiguation using Machine Readable Dictionaries: How to Tell a Pine Cone from an Ice Cream Cone." in *Proceedings of ACM SIGDOC Conference*, Toronto, Canada, 1986 p. 25-26

[9] G. Miller and R. Beckwith and C. Fellbaum and D. Gross and K. Miller Introduction to Wordnet: an Online Lexical Database. in *International Journal of Lexicography*, 1993 p. 234 244.

[10] S. Banerjee and T. Pedersen "An Adapted Lesk Algorithm for Word Sense Disambiguation Using WordNet." in *Lecture Notes In Computer Science*, Springer, 2002.

[11] G. Hirst *Semantic Interpretation and the Resolution of Ambiguity*, Cambridge University Press, 1987.

[12] H. Ng and H. Lee "Integrating multiple knowledge sources to disambiguate word sense: An examplar-based approach." *Proceedings of the 34th Meeting of the Association for Computational Linguistics (ACL-96)*, Washington, DC, 1996, pp. 40-47.

[13] D. Yarowsky "Word Sense Disambiguation" in N. Indurkhya and F. Damerau *Handbook of Natural Language Processing - 2nd Edition*, Chapman and Hall, Florida, 2010, pp. 315 338.

[14] H. Schtze "Automatic word sense discrimination". *Computational Linguistics 24*, 1988, pp. 97 124.

[15] R. Navigli and M. Lapata "Graph connectivity measures for unsupervised word sense disambiguation." *Proceedings of IJCAI*, Hyderabad, India, 2007, pp. 1683 1688.

[16] W. Chang and J. Preiss and M. Stevenson "Scaling up WSD with Automatically Generated Examples." *Proceedings of the 2012 Workshop on Biomedical Natural Language Processing (BioNLP 2012)*, Montreal, Canada, 2012, pp. 231239.

[17] M. Zampieri "A Supervised Machine Learning Method for Word Sense Disambiguation of Portuguese Nouns." *Bulletin de Linguistique Aplique et Gnrale - BULAG 34*, Besanon, France, 2010, pp. 187 - 203.

[18] H. Schmid "Probabilistic Part-of-Speech Tagging Using Decision Trees." *Proceedings of International Conference on New Methods in Language Processing*, Manchester, UK, 1994.

[19] L. Specia *Uma abordagem hibrida relacional para a desambiguacao lexical de sentido na traducao automatica*, PhD Thesis, 2007.

[20] J. Baptista and N. Costa and J.Guerra and M. Zampieri and M. Cabral and N. Mamede "P-AWL: Academic Word List for Portuguese." *PROPOR2010, Lecture Notes in Artificial Intelligence LNAI 6001*, 2010, pp. 120-123.

[21] A. Coxhead "A New Academic Word List", *TESOL Quarterly 34*, 2010, pp. 213238.

[22] R. Koeling and D. McCarthy and J. Carroll "Text categorization for improved priors of word meaning." *Proceedings of the Eighth International Conference on Intelligent Text Processing and Computational Linguistics (CICLING 2007)*, Mexico City, Mexico, 2007.

[23] H. Schmid "Probabilistic Part-of-Speech Tagging Using Decision Trees." *Proceedings of International Conference on New Methods in Language Processing.* Manchester, UK, 1994.

[24] S. Bird and E. Klein and E. Loper *Natural Language Processing with Python Analyzing Text with the Natural Language Toolkit*, OReilly Media. 2009.

[25] I. Witten and E. Frank *"Data Mining: Practical Machine Learning Tools and Techniques (2nd Edition)*, Morgan Kaufmann, San Francisco. 2005.

[26] J. Cleary and L. Trigg "K*: An instance-based learner using an entropic distance measure". In: A. Prieditis and S. Russell (eds.). *Proceedings of the Twelfth International Conference on Machine Learning*, Tahoe City, Ca. Morgan Kaufmann, p.108-114. 1995