Word Sense Disambiguation
with the KORA-Ω Algorithm

Abstract. We present a method for word sense disambiguation (WSD) based on the KORA-Ω supervised learning algorithm. The advantage of the method is its simplicity and a very small feature set used, though, as we show, this is achieved at the cost of lower accuracy of the final result than the complex state-of-the-art methods achieve.

1   Introduction

Word Sense Disambiguation (WSD) is the processes of selecting the most appropriate meaning for a word based on the context in which it occurs. For example, in the phrase *The bank down the street was robbed*, the word *bank* means a financial institution, while in *The city is on the Western bank of Jordan*, this word refers to the shore of a river. The WSD is an intermediate task [17] in natural language processing chain, essential for applications such as information retrieval or machine translation.

It can be thought of as a classification task, where word senses are the classes, the context provides the evidence, and each occurrence of a word is assigned to one of its possible classes based on the evidence. This task is often treated as a supervised learning problem, where a classifier is trained from a corpus of manually sense-tagged texts using machine learning methods. These approaches typically represent the context in which each sense-tagged instance of the ambiguous word occurs using the features such as the part-of-speech (PoS) of surrounding words, keywords, syntactic relationships, etc.

To address this task, different statistical methods have been proposed, with various degrees of success. This includes a number of different classifiers like Naïve Bayes [12], neural networks [5], and content vector-based classifiers [8].

In this paper we present a supervised learning method based on the Logical Combinatorial Pattern Recognition (LCPR) approach, called KORA-Ω [4].

The paper begins with a review of the related works in the area. In Section 3 the proposed method is presented. Section 4 reports our experimental results. Section 5 concludes the paper and introduces some future work.

2   Related Work

Several research projects take a supervised learning approach to WSD [3, 6, 8]. The goal is to learn to use surrounding context to determine the sense of an ambiguous word.

Often the disambiguation accuracy is strongly affected by the size of the corpus used in the process. Typically, 1000–2500 occurrences of each word are manually tagged in order to create a corpus. From this about 75% of the occurrences are use for the training phase and the remaining 25% are use for the testing [11]. Corpus like interest and line were the most well studied in literature.

The *Interest* dataset (a corpus where each occurrence of the word interest is manually marked up with one of its 6 senses) was included in a study by [13], who represent the context of an ambiguous word with the part-of-speech of three words to the left and right of interest, a morphological feature indicating if interest is singular or plural, an unordered set of frequently occurring keywords that surround interest, local collocations that include interest, and verb-object syntactic relationships. A nearest-neighbor classifier was employed and achieved an accuracy of 87% over repeated trials using randomly training and test sets. Ng and Lee [7], and Pedersen et al. [15] present studies that utilize the original Bruce and Wiebe feature set and include the interest data .The first compares a range of probabilistic model selection methodologies and finds that none out perform the Naive Bayesian classifier, which attains accuracy of 74%. The second compares a range of machine learning algorithms and finds that a decision tree learner 78% and a Naïve Bayesian classifier 74% are most accurate.

The *Line* dataset (similarly, a corpus where each occurrence of the word *line* is marked with one of its 6 senses) was first studied by Leacock [8]. They evaluate the disambiguation accuracy of a Naive Bayesian classifier, a content vector, and a neural network. The context of an ambiguous word is represented by a bag-of-words (BoW) where the window of context is two sentences wide. When the Naive Bayesian classifier is evaluated words are not stemmed and capitalization remains. With the content vector and the neural network words are stemmed and words from a stop-list are removed. They report no significant differences in accuracy among the three approaches; the Naïve Bayesian classifier achieved 71% accuracy, the content vector 72%, and the neural network 76%.

This dataset was studied again by Mooney [12], where seven different machine learning methodologies are compared. All learning algorithms represent the context of an ambiguous word using the BoW with a two sentence window of context. In these experiments words from a stop list are removed, capitalization is ignored, and words are stemmed. The two most accurate methods in this study proved to be a Naive Bayesian classifier 72% and a perceptron 71%.

Recently, the Line dataset was revisited by both Towell and Voorhees [5], and Pedersen [14]. Take an ensemble approach where the output from two neural networks is combined; one network is based on a representation of local context while the other represents topical context. The latter utilize a Naive Bayesian classifier. In both cases context is represented by a set of topical and local features. The topical features correspond to the open-class words that occur in a two sentence window of context. The local features occur within a window of context three words to the left and right of the ambiguous word and include co-occurrence features as well as the PoS of words in

this window. These features are represented as local and topical BoW and PoS. [5] report accuracy of 87% while [15] report accuracy of 84%.

3   Proposed Method

The KORA-Ω algorithm is an extension of the widely used KORA-3 [2] in geosciences. This algorithm was used for supervised classification problems. The algorithm works with disjointed classes and objects. The idea is to classify new objects (patterns) based on a training sample of objects through the verification of some complex properties. These complex properties are a combination of certain feature values, named complex features (CF) that discriminate an object or a set of objects in the same class from the remaining objects in different classes. Fuzzy KORA-Ω allows us to solve supervised classification problems with many classes (hard-disjointed or fuzzy), with any kind of features. In this model, complex properties could be of any length greater or equal to one.

The algorithm works based on the idea of finding for each object of the training matrix a property such there is a few other identical property in any object of the remaining classes. In general, we can describe this family of algorithms in three stages:

Learning stage   It is necessary all the parameters in order to determine which property of feature values are complex features for each class. The entire objects in each class are covered for enough complex features. Enough is also a parameter of the algorithm. All the classified objects in each class satisfy at least a predetermined number of complex features.

Relearning stage   In this second stage the problem is the same as in the previous one, but with other parameters, shorter than the "enough" of the previous stage.

Classification stage   Finally, we have all the complex features for each class, with different levels of discrimination (from the learning stage and the relearning stage). If a given complex feature for a class is present in the new object to classify, this class receives a vote. After that, any decision-making rule is applied.

3.1   Algorithm

The disambiguation of a particular word W is performed as follows:

INPUT: semantically untagged pattern of W and its context.
OUTPUT: semantically tagged pattern of W and its context.

Learning Stage

Step 1. Define β_1 and β_2 thresholds
Step 2. Define the properties.
Step 3. Find the characteristic CF, β_1-caracteristic.

Relearning stage

Step 4. Calculate the rest of the class, using the definition of complement.
Step 5. Define β_3 and β_4 thresholds
Step 6. Find the complementary CF, β_3-complementary

Classification Stage

Step 7. Apply a decision-making rule.

3.2   Decision-making Rule

The final step of the classification stage is divided in 3 sub-steps:

Calculate the voting scheme. If an object fulfill with an a characteristic CF, then it gives one vote. For a complementary CF the process is the same. After a counting of the CF's we weight the characteristics voting with 0.7 and the complementary with 0.3. Finally the vote of the class is the sum of both characteristic and complementary CF.

Class membership. An pattern is assigned to the class with the biggest sum of CF.

Amount of membership. The pattern is assigned with membership degree 1 to the class of the previous step and 0 to the others.

4 Experimental Results

4.1 Data Set

The Line dataset was developed for the task of disambiguation of the word *line* into one of six possible senses (text, formation, division, phone, cord, product) based on the words occurring in the current and previous sentence. The corpus was assembled from the 1987-89 Wall Street Journal and 25 million word corpus from the American Printing House for the Blind. Sentence containing *line* were extracted and assigned a single sense from WordNet [9]. There are a total of 4,149 examples in the full corpus unequally distributed across the six senses. This dataset and distribution of senses are shown in Table 1.

In this work, we used a subset of the Line dataset in which every sense is equally distributed taking 349 sense-tagged examples for each sense resulting in a training corpus of 2094 sense-tagged sentences. We form every sentence in a pattern format using only 3 open-class words to the left and right around the ambiguous word and leaving only the word an each PoS. We use for tuning the thresholds $\beta_1 = 3$, $\beta_2 = 0$, $\beta_3 = 2$, and $\beta_4 = 0$, see Section 3.

Table 1. Distribution of senses in Line dataset.

Sense	count
Product	2218
written or spoken text	404
telephone connection	429
formation of people or things; queue	349
an artificial division; boundary	376
a thin, flexible object; cord	373
Total:	4149

4.2 Complex Features

The local context is a window of lexical units that occur around the ambiguous word, varies from few words to the entire sentence. Some parameters that have been used are: distance, collocation, and syntactic information.

The concept of distance is related with the number of words (n) in the context. Studies gave different answers for an optimal number of n, Ide and Veronis [10], have shown that 2 words are enough for the WSD task, even 1 word is trustful. Other studies [18], reach the conclusion of an optimal n value for a local context in 3 or 4. Using in this paper as feature set of 3 words around the ambiguous word to the right and left. Creating 9 complex features sets, where are the combinations of words, and

other sets for the PoS. For example the set $CF_1 = \{P:0, P:+1\}$, where P is the word and the number is the position in context.

4.3 Comparison with Previous Results

The best result of our method was achieved by using the complex features of words with a decreasing of accuracy using only the PoS complex features. Table 2 shows the accuracy compared to other methods, as evaluated in the Line dataset.

Table 2. Comparasion with previous results.

Method	Accuracy	Algorithm	Feature set
Pedersen 2000	88%	Naive Bayesian Ensemble	varying left & right; BoW
Towell & Voorhess 1998	87%	Neural network	local & topical BoW; PoS
Leacock, Chodorow & Miller 1998	84%	Naive bayes	local & topical BoW; PoS
Leacock Towell & Voorhees 1993	76% 72% 71%	Neural network Content vector Naive bayes	2 sentence BoW
Mooney 1996	72% 71%	Naive bayes Perceptron	2 sentence BoW
Proposed	60% 53%	KORA-Ω KORA-Ω	3 Word properties 3 PoS properties

Our algorithm uses a very limited feature set, even though at the cost of lower results as compared to complex state-of-the-art techniques. We believe that the algorithm could give better results by using more information of the context—for example, a wider window for both the words and PoSs or the use of information other than lexical, e.g., morphological. However, with this our algorithm would possibly lose its main advantage: simplicity.

Another possible improvement for the method is selecting of less restrictive complex features and thresholds such as β_2 and β_4, to permit any repetition of a complex feature in the other classes—things that KORA-3 does not allow.

5 Conclusions and Future Work

In this paper we used KORA-Ω algorithm for the WSD task. This algorithm has the advantage of simplicity and the use of a very limited feature set, though at the cost of the accuracy of the final result.

Essentially, the KORA-Ω makes a positive characterization of a class (properties which belongs to the class) based on the idea of majorities and minorities of the population. For future work we will try to also make negative characterization of a class: properties which no belong to the class, as the Representative Set algorithms do [1].

We also plan to experiment with adding a limited subset of linguistically-motivated features, as well to try wider windows and more open thresholds.

References

1. Baskakova, L.V., and Yu.I. Zhuravlev. 1981. Model of algorithm of recognition with sets and support sets systems. Journal Zhurnal Vichislitielnoi Matemati y Matematicheskoi Fisiki 21, No. 5, 1264.
2. Bongard, M. N. et al. 1963. Solving geological problems using recognition programs. Journal Soviet Geology, 6C. Leacock, M. Chodorow, and G. Miller. 1998. Using corpus statistics and WordNet relations for sense identification. Computational Linguistics, 24(1):147–165, March.
3. Brown, P., Della-Pietra, S., Della-Pietra, V., & Mercer, R. 1991. Word sense disambiguation using statistical methods. In Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics, pp. 264–270.
4. De-la-Vega-Doria, L., J.A., Ruiz-Shulcloper, J., and Carrasco-Ochoa. 1998. Fuzzy KORA-Ω algorithm. Proceeding of the Sixth European Congress on Intelligent Techniques and Soft Computing, EFIT'98, 1190–1194. Aachen, Germany
5. G.Towell and E.Voorhees.1998.Disambiguating highly ambiguous words. Computational Linguistics, 24(1):125–146, March.
6. Gale, W., Church, K., & Yarowsky, D. 1992. A method for disambiguating word senses in a large corpus. Computers and the Humanities,26, 415–439.
7. H.T. Ng and H.B. Lee. 1996. Integrating multiple knowledge sources to disambiguate word sense :An exemplar-based approach. In Proceedings of the 34th Annual Meeting of The Society for Computational Linguistics, pages 40–47.
8. Leacock, C., Towell, G., & Voorhees, E. 1993.Corpus-based statistical sense resolution. In Proceedings of the ARPA Workshop on Human Language Technology.
9. Miller, G. 1991. WordNet: An on-line lexical database. International Journal of Lexicography, 3(4).
10. Nancy Ide and Jean Véronis.1998.Word sense disambiguation: The state of the art. Computational Linguistics.
11. Rada Mihalcea and Dan Moldovan. 1999. An Automatic Method for Generating Sense Tagged Corpora, in Proceedings of the American Association for Artificial Intelligence, Orlando, FL, July.
12. R. Mooney. 1996. Comparative experiments on disambiguating word senses: An illustration of the role of bias in machine learning. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, pages 82–91.
13. R. Bruce and J. Wiebe. 1994. Word-sense disambiguation using decomposable models. In Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics, pages 139–146.
14. Ted Pedersen. 2000. A simple approach to building ensembles of Naive Bayesian classifiers for word sense disambiguation. Proceedings of the first conference on North American chapter of the Association for Computational Linguistics. Seattle, Washington, pages 63–69.

15. T. Pedersen, R. Bruce, and J. Wiebe. 1997. Sequential model selection for word sense disambiguation. In Proceedings of the Fifth Conference on Applied Natural Language Processing ,pages 388–395,Washington, DC, April.

16. T. Pedersen and R. Bruce. 1997. A new supervised learning algorithm for word sense disambiguation. In Proceedings of the Fourteenth National Conference on Artificial Intelligence, pages 604–609.

17. Wilks, Yorick and Stevenson, Mark . 1996. The grammar of sense: Is word sense tagging much more than part-of speech tagging? .Technical Report CS-96-05, University of Sheffield, Sheffield, United Kingdom.

18. Yarowsky, David. 1994. "Decision Lists for Lexical Ambiguity Resolution: Application to Accent Restoration in Spanish and French," in Proceedings of the 32nd Annual Meeting of the Association .for Computational Linguistics, Las Cruces, NM.