International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2015): 6.391

Supervised Word Sense Disambiguation

Mihir Sawant[1], Tanya Sangoi[2], Sindhu Nair[3]

[1]Student, D.J. Sanghvi College of Engineering

[2]Student, D. J. Sanghvi College of Engineering

[3]Assistant Professor, D. J. Sanghvi College of Engineering

Abstract: ***Word Sense Disambiguation (WSD) is the method of the correct sense for word in a context. In this paper we have researched the various approaches for WSD: Knowledge based, Supervised, Semi-supervised, Unsupervised methods. This paper has further elaborated on the supervised methods used for WSD. The methods that are compared in this paper are: Decision Trees, Decision Lists, Support Vector Machines, Neural Networks, Naïve Bayes methods, Exemplar learning.***

Keywords: Word Sense Disambiguation, Natural Language Processing, Supervised Learning, Knowledge Acquisition Bottleneck, Sense tagged corpora.

1. Introduction

In every language, there are several words which have multiple meanings that change depending on the context in which they are used. Word Sense Disambiguation is used to remove such ambiguities that occur in a given context. For example, the word "bark", it can mean either the outer covering of a tree, as a noun, or it can mean the sound made by a dog, as a verb, depending on the context. Human beings possess the innate capability to differentiate between multiple such meanings of a word; however, machines need to be instructed to perform this particular task. To disambiguate a word, a dictionary is required to specify the senses and a corpus is used to provide a context. WSD is classified as an AI-complete problem which means that its solution is at least as hard as the most difficult problems in Artificial Intelligence.

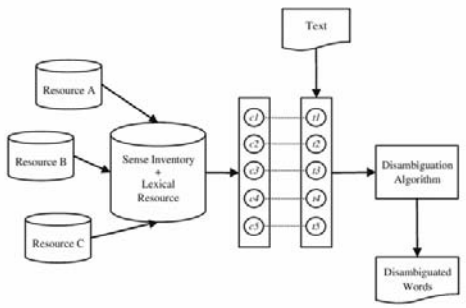

Figure 1: General Model of WSD [1]

There are several important applications of WSD:

1) *Information Retrieval*: WSD helps in retrieving of the best results by minimizing the ambiguity of the query or documents translation.
2) *Machine Translation*: It is used to remove ambiguity between different senses of a word in a given domain to convey information to a machine correctly and provide further conversion using intermediate code.
3) *Information Extraction*: This focuses on extracting certain information from a single document. For example, when searching for an article on the internet, there may be results that don't match the user requirements or there may be too many similar articles in the search results. In this case, WSD is used to remove confusion and find out the correct sense of the word.
4) *Speech Processing*: WSD is used in speech processing to distinguish between similar sounding words which have different meanings, for instance, "write" and "right".[2]

2. Approaches to WSD

The approaches to WSD are as follows:

A. Knowledge Based Approach

Knowledge based methods are a distinct type of WSD algorithm which came out into existence in the 1970's and 1980's.[3] These algorithms avoid the need for large amounts of training material and exploit the knowledge contained in several resources like dictionaries, thesauri, WordNet, Stemcor, Wikipedia etc. to provide the appropriate sense of word in a context.

Such algorithms are developed for automatic sense tagging and they can be used for all words in an unrestricted text. [4].These algorithms have an advantage over corpus based algorithms which are only applicable for those words for which annotated corpora are available. However corpus based algorithms are more precise than knowledge based ones.

The various types of knowledge based approaches are:

1) Overlap of Sense Definitions
2) Selectional Preferences
3) Structural Approaches

B. Supervised Approach

Supervised WSD uses Machine learning techniques are used to perform WSD. A classifier is assigned to a single word and is used to assign the appropriate sense to each instance of that word. Supervised approach requires manually created training data followed by testing phase in which classifiers try to find the most suitable sense of the word in a context. Generally, supervised approaches are more accurate than the other approaches.

C. Semi-supervised Approach

In this approach, both annotated and unannotated data is used. Bootstrapping algorithm was the first semi-supervised

Licensed Under Creative Commons Attribution CC BY

algorithm. It involves using a small set of annotated data, a larger set of unannotated data and a set of classifiers. The algorithm is then applied over both of them, which results in the annotated dataset expanding while the unannotated set shrinks till some threshold is reached. High accuracies have been observed when this algorithm is applied on a smaller dataset [5]. A drawback here is of the number of uncertainties involved while selecting parameter values like pool size and number of iterations [6].

D. Unsupervised Approach

Supervised approaches have been found to be largely superior to unsupervised approaches. However, they require large amounts of data to be trained and their scope is limited to words for which the senses are labeled. Therefore, in their purest version, they do not rely on external sources of knowledge, sense inventories or machine readable dictionaries. This problem is referred to as Knowledge Acquisition Bottleneck.

The primary task of unsupervised WSD approaches is that they aim to identify sense clusters as opposed to assigning sense labels like in supervised methods. There are several methods involved here: context clustering, word clustering and co-occurrence graph [5]. A few limitations to such an approach exist. It is not suitable for a larger scale situation, incorrect assignment of instances in training data, formation of heterogeneous clusters and the difference between the number of clusters formed and the number of senses of the target word.

The various types of unsupervised based approaches are:
1) Context Clustering
2) Word Clustering
3) Co-occurrence Graphs

3. Types of Supervised Approaches

A. Decision tree

In the decision tree approach a sense tagged corpus is used as a resource to perform training. A classification rule is applied in the form of "yes-no" rule is used to recursively divide the training data set. [7] A test which is going to be applied on a feature value is represented with the internal node of a decision tree and the output is denoted by every. The sense of the word is represented when the leaf node is reached.

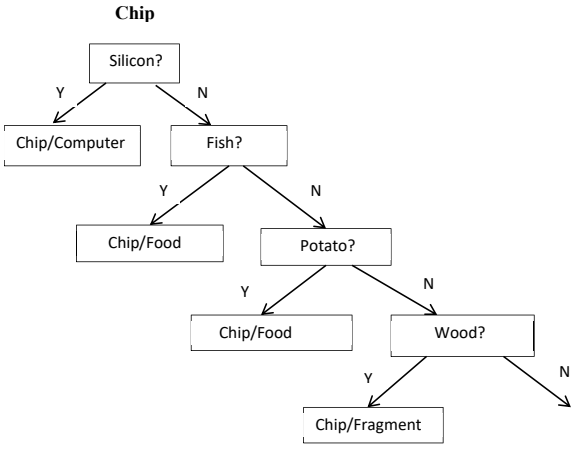

Figure 2: An example of a Decision tree

B. Decision Lists

Decision Lists are an ordered set of rules for assigning the appropriate sense of the word. It is a list of weighted "if-then-else" rules. [8] To induce the set of features for a given word, training sets are used. Few parameters like sense, feature-value, and score are created using those rules [5]. The ordering of these rules, based on their decreasing score, constitutes the decision list. The decision list is checked, given the word w and its representation as a feature vector. The feature with the highest score that matches the input vector selects the word sense to be assigned.[5]

$$\hat{S} = argmax S_i \in SensesD(w)\ score(S_i).$$

According to Yarowsky [1994], the score of sense Si is calculated as the maximum among the feature scores, where the score of a feature f is calculated as the log of the probability of sense Si given feature f divided by the sum of the probabilities of the other senses given feature f [5][9]:

$$score(S_i) = max_f log\left[\frac{P(Si \mid f)}{\sum_{j \neq i} P(Sj \mid f)}\right]$$

C. Support Vector Machines

SVMs are linear classifier that produces the hyperplane for separating the positive and negative training examples with largest margin, where margin is the distance of hyperplane to the nearest of the positive and negative examples. The examples which are closest to the hyperplane are called support vector. In order to be usable for WSD, as a binary classifier, a SVM must be adapted to multiclass classification (i.e., the senses of a target word). The test example is classified depending on the side of the hyperplane it lies on[10]. Kernel functions are used to reduce the computational cost of the training and testing procedures in high dimensional space. The default linear kernel is used.

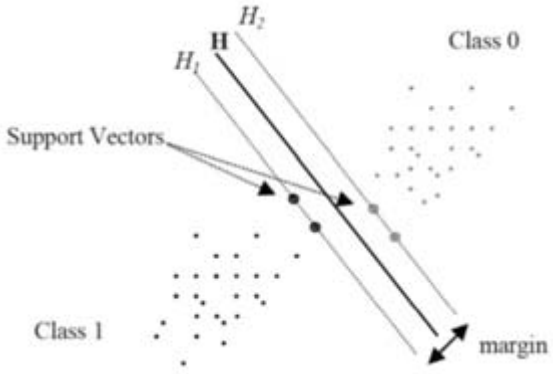

Figure 3: Support Vector Machine [11]

D. Neural Networks

Neural Network model consists of interconnected group of artificial neurons which are used for data processing. The pairs of (input features, desired responses) are the input of this learning program and the goal is to partition the training context into non-overlapping sets[8]. The inputs are propagated from the input layer to the output layer through the all intermediate layers. As new pairs are provided, link weights are progressively adjusted so that the desired response which is represented bythe output unit has a larger activation than any other output unit[9]. The major problems

Licensed Under Creative Commons Attribution CC BY

that occur in neural networks are: difficulties in interpreting the results, the need for a large quantity of training data, and the tuning of parameters.

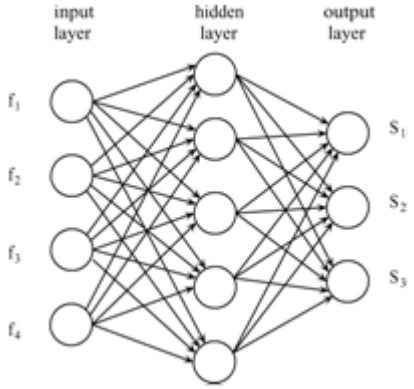

Figure 4: A Multilayer Neural Network [5]

E. Naïve Bayes Method

The Naïve Bayes classification method was first used for WSD by Gale et al. (1992). This is a probabilistic approach that works on the basis of Bayes theorem. It is assumed that the feature variables representing a problem are conditionally independent, given the classes. The conditional probability is calculated for each sense (k) of a word (w) given the context C and features F = (f_1, f_2 . . . f_n):

$$k = arg_{s_i} \max P(w = s_i | F)$$
$$= \arg_{s_i} max \frac{P(F \mid w = s_i)}{P(F)} P(w = s_i)$$
$$= \arg_{s_i} \max P(F \mid w = s_i) P(w = s_i)$$

P(w = s_i) and P(x_i/k) are the probabilistic parameters of the model and they can be estimated from the training set, using relative frequency counts.

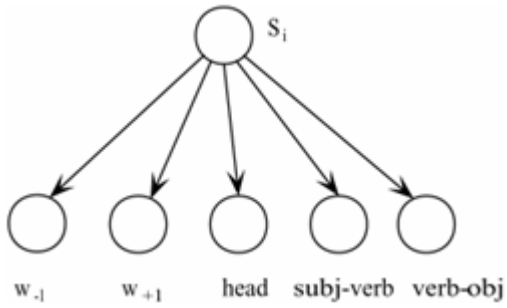

Figure 5: An example of a Bayesian network [5]

The figure shows a simple example of a Bayesian network. For example, in a sentence, the boy caught the ball, if we want to classify the occurrence of the noun boy, the features will be recognised as follows: {w-1 = the, w+1 = caught, head = boy, subj-verb = catch, verb-obj = -}. From the training set, the probability of these features will be determined based on the desired classification of the word "boy". The final score is then obtained by finding the product of all these probabilities.

F. Exemplar-based Method

Exemplar-based learning involves retaining examples in memory as points in a feature space. New examples are then individually added to the feature space. The k-Nearest Neighbor algorithm is based on this approach. This is one of the highest performing models in WSD [12].

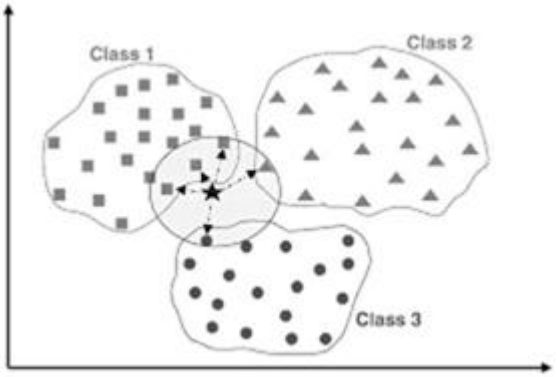

Figure 6: An example of kNN classification on a bidimensional plane [13].

For every new example $x = (x_1, \ldots, x_m)$ represented in terms of its m feature values, its classification is based on the previously stored k nearest examples in the feature space. The distance between x and the stored examples $x_i = (x_{i_1}, \ldots, x_{i_m})$ is calculated by the Hamming distance [5],

$$d = \sum_{j=1}^{m} w_j \, \delta\left(x_j, x_{i_j}\right),$$

Where d is the distance between x and x_i, w_j is the weight of the j th feature and $\delta(x_j, x_{i_j})$ is 0 if $x_j = x_{i_j}$ and 1 otherwise. Then, the set of the closest examples is examined and the new sense of the word is predicted by the classification of the majority of the nearest examples. At present, exemplar-based learning achieves state-of-the-art performance in WSD.

4. Conclusion

Word Sense Disambiguation, being an AI-complete problem, is one of the hardest NLP problems. This work explores approaches to tackle WSD using manually created training data. Supervised techniques such as, Decision Lists, Decision Trees, Naive Bayes classification, Support Vector Machines, Neural Networks and Exemplar Learning are further studied. While supervised methods are generally superior to other approaches to WSD, there are several drawbacks to the same. These techniques rely on large amounts of manually sense-tagged corpora for training which are arduous and expensive to create. While, supervised methods show great results in related domains, a "knowledge acquisition bottleneck" is experienced due to the lack of widely available semantic-tagged data. There is an extremely high overhead for supervision since it would require approximately 16 man-years to create a broad coverage semantically annotated corpus. Furthermore, when common Machine Learning models scale to real-size WSD problems, a serious learning overhead occurs

References

[1] New Perspectives on Computational and Cognitive Strategies for Word Sense Disambiguation. New York, NY: Springer New York, 2013.

Licensed Under Creative Commons Attribution CC BY

[2] G. Chandra and S. Dwivedi, "A Literature Survey on Various Approaches of Word Sense Disambiguation", 2014 2nd International Symposium on Computational and Business Intelligence, 2014.
[3] Knowledge-based WSD", www.cs.cmu.edu, 2016. [Online]. Available: https://www.cs.cmu.edu/afs/cs/project/jair/pub/volume23/montoyo05a-html/node5.html. [Accessed: 20- Oct- 2016].
[4] E. Agirre and P. Edmonds, Word sense disambiguation. Dordrecht: Springer, 2006.
[5] R. Navigli, "Word sense disambiguation", ACM Computing Surveys, vol. 41, no. 2, pp. 1-69, 2009.
[6] V. Ng and C. Cardie, "Weakly supervised natural language learning without redundant views", in Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics - Volume 1, 2003, pp. 173-180.
[7] D. Sreenivasan and V. M, "A Walk Through the Approaches of Word Sense Disambiguation", International Journal for Innovative Research in Science & Technology, vol. 2, no. 10, p. 041, 2016.
[8] A. Ranjan Pal and D. Saha, "Word Sense Disambiguation: A Survey", IJCTCM, vol. 5, no. 3, pp. 1-16, 2015.
[9] N. Patel, B. Patel, R. Parikh and B. Bhatt, "A Survey: Word Sense Disambiguation", International Journal of Advance Foundation and Research in Computer, vol. 2, no., 2015.
[10] Y. Lee, H. Ng and T. Chia, "Supervised Word Sense Disambiguation with Support Vector Machines and Multiple Knowledge Sources", in SENSEVAL-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text, Barcelona, Spain, 2004.
[11] J. R., M. del, M. A., W. G., C. M. and A. B., "Image Processing for Spider Classification", Biodiversity Conservation and Utilization in a Diverse World, 2012.
[12] T. Ng, "Exemplar-Base Word Sense Disambiguation: Some Recent Improvements", in Proceedings of the 2nd Conference on Empirical Methods in Natural Language Processing, EMNLP, 1997.
[13] W. Chaovalitwongse, Y. Fan and R. Sachdeo, "On the Time Series K-Nearest Neighbor Classification of Abnormal Brain Activity", IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans, vol. 37, no. 6, pp. 1005-1016, 2007

Licensed Under Creative Commons Attribution CC BY