International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2013): 4.438

Supervised, Semi-Supervised and Unsupervised WSD Approaches: An Overview

Lokesh Nandanwar[1], Kalyani Mamulkar[2]

[1, 2]Yeshwantrao Chavan College of Engineering, RTM Nagpur University, Nagpur, Maharashtra, India

Abstract: ***Word Sense Disambiguation (WSD) involves the identification of a correct sense of a word in a given sentence. WSD is considered to be an open and AI-complete problem of Natural Language Processing (NLP). WSD is found to be most important in many applications like Machine translation (MT), Information retrieval (IR), Information extraction (IE), text mining, and Lexicography. Supervised, Semi-supervised and Unsupervised Approaches to WSD are found to be important and very successful learning approaches. These methods are categorized based on the main source of knowledge used to differentiate senses or type and amount of annotated (labeled) corpora (data) required. Semi-supervised approach requires lesser quantity of annotated corpora as compared to supervised approaches which needs large amount of annotated corpora while unsupervised approach uses unannotated (unlabeled) corpora for training. In this paper, we will discuss all the three approaches and their respective methods in details.***

Keywords: Word Sense Disambiguation, Natural language processing, Supervised approach, Semi-supervised approach, unsupervised approach.

1. Introduction

In almost all languages, lexical ambiguity is a fundamental characteristic. A word may have more than one different senses or meanings and then that word is considered to be ambiguous word. Resolving such an ambiguity of a word is called as Word Sense Disambiguation [1]. Therefore WSD is the task of identifying the correct sense (meaning) of a word and replacing it by the same in a given context. For example, consider a word *grain* in English which has two meanings one as noun: a small hard seed of a cereal plant such as wheat and the other as noun: the lines made by fibres in wood, or texture in a fabric. Word sense disambiguation replaces the ambiguous word by the proper one depending on the surrounding context [4].

The word sense disambiguation can be easily achieved by using knowledge based trained data and feature selection. Knowledge based trained data can be unlabelled or annotated with word senses and are vary from one another depending on applications. For example, data of medical field and administrative field. WSD is first developed to achieve the aim of easy machine translation (MT) in the field of computational linguistics. By the time many methods are proposed to resolve the ambiguity which includes Supervised learning algorithms (Leacock et al., 1998), Semi-supervised learning algorithms (Yarowsky, 1995), & Unsupervised learning algorithms (Schutze, 1998).

In supervised approach the algorithms works on already trained or classified (sense-tagged) data which can be in the form of wordnets [2] or knowledge bases to differentiate the new data. Supervised learning approach requires the large amount of labeled data for training to achieve the good performance. In case of semi-supervised approach the training data can be labeled or unlabeled or partially trained data. While the unsupervised learning approach works on raw data which is not sense tagged so that clustering methods are used. The continuous research is going on semi-supervised and unsupervised approaches to achieve state of art performance. This paper is mainly organized as follows: Section 2 includes supervised approach, Section 3 includes semi-supervised approach and Section 4 includes semi-supervised approach.

2. Supervised Approach

Supervised algorithm uses sense-annotated trained corpora to distinguish the senses of the words. The supervised approach [3] uses two phases namely training and testing phases. Training phase requires a sense-annotated training corpus to built classifiers using machine learning techniques. While in the testing phase classifiers tries to recognize the required senses depending on surrounding sentence. There are number of classifiers presents also called as *word experts* that are used to classify the appropriate meaning of a single word. Supervised approach always gives better performance than any other methods.

There are various supervised methods that are available viz. method based on similarity measures [7], probabilistic methods, methods based on discriminating rule and methods based on linear classifications. In similarity based methods the disambiguation is done by comparing the features of new or raw sample data with the features of trained sample data and assign the sense of most similar pattern. Probabilistic methods estimates set of parameter such as conditional or joint probability distribution. In discriminating rule methods, some rules are used which are associated with each word sense, to classify new sample one or more rules are selected that satisfy sample feature and assign sense based on their predictions.

Each and all supervised algorithm uses certain features associated with a sense for training which is common thread of functionality of supervised algorithms. In this section we will discuss some important supervised algorithms for word sense disambiguation.

2.1 Naïve Bayes Method

Naïve Bayes Method is considered to be supervised method. This method makes use of probabilistic approach which is one of the statistical methods, used to estimate probabilistic parameters. This probabilistic approach usually expresses joint probability distribution or conditional probabilities in a given context and categories. Naïve Bayes algorithm [13] uses classifiers which are mainly based on Bayes theorems to calculate the conditional probability for each sense (say k) of a word for which the features are defined (x_1, x_2, …, x_m). Let P(k) 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.

$$\arg\max_k P(k \mid x_1,\ldots,x_m) = \arg\max_k \frac{P(x_1,\ldots,x_m \mid k)P(k)}{P(x_1,\ldots,x_m)}$$

$$= \arg\max_k P(k)\prod_{i=1}^{m} P(x_i \mid k).$$

2.2 Decision List Method

Another supervised method which uses ordered set of rules for categorizing test instances is called a decision list method [5]. In decision list method weighted 'if then else' rules are used. The method considers following important features for each word which include collocation vector and co-occurrence vector, syntactic and semantic feature, part-of-speech. The training labeled data set is used to train the classifiers for the first time and which needs to identify the important features. The predicate rules are created in the form (feature-value, sense, score). Then these rules are sorted in non-increasing order and thus formation of decision list takes place. While testing, the decision list is scanned for the entries which matches input feature vector, the sense with highest score will select as the accurate sense and thus the word is disambiguated.

$$\text{weight}(s_k, f_i) = \log\left(P(s_k \mid f_i) \Big/ \sum_{j \neq k} P(s_j \mid f_i)\right)$$

2.3 Decision Tree Method

The Supervised decision tree method is based on the prediction based model. The sense tagged corpus is used as knowledge source on which the training is to be done. The yes-no form of rules are used as classification rules in this method. This rules are then used to recursively parsed training data. The main feature of this method is that all the internal nodes are used to represent the features while each edge are representing feature values and all leaf nodes are used to represent the important senses. The feature vectors in both the cases i.e. in decision tree method and in the decision list methods are same. While testing, the ambiguous word with corresponding feature vector is traversed throughout to reach to leaf node. Then a reached leaf node sense is considered to be correct sense of the ambiguous word.

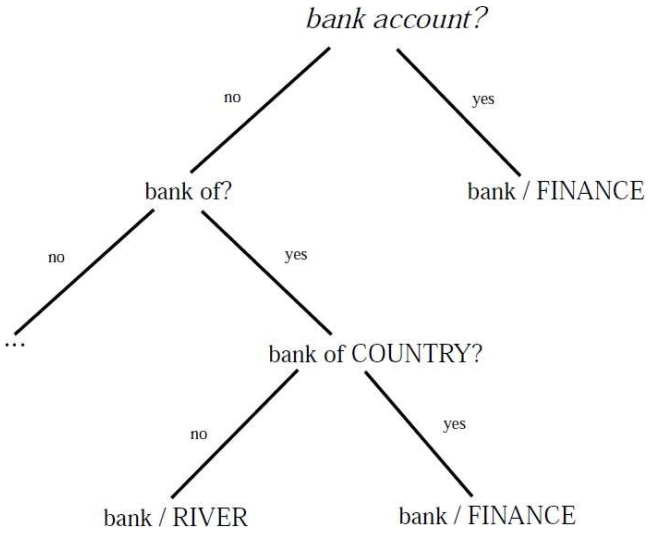

Figure 1: Example of Decision Tree

2.4 Support Vector Machine (SVM) Method

Support Vector Machine method is one of the supervised method that separates positive samples from negative samples. It is basically based on the idea of linear hyper-plane from labeled data set. This method uses SVM binary classifier to differentiate between samples into either true or false category. SVM is adapted to multicast classification for word sense disambiguation. It is then converted into binary classification problem of the kind sense Si versus all other senses.

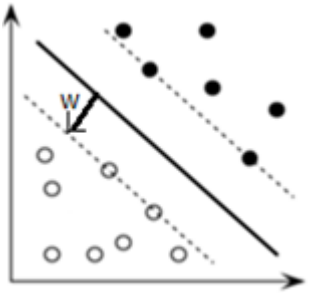

Figure 1: Example of SVM method

2.5 Exemplar-Based Learning Method

This supervised learning method uses the memory to store the training data. This method is based on the learning method k-Nearest Neighbor (kNN) algorithm to classify testing data based on the senses of k most similar stored examples. The set of nearest neighbor is obtained by comparing each feature of testing data x = (x_1, …, x_m) with respective feature of each training data set x^i = (x^i_1, …, x^i_m). Then the distance between them is calculated using hamming distance method as follows:

$$\Delta(x, x^i) = \sum_{j=1}^{m} w_j\, \delta(x_j, x_j^i)$$

Where, w_j is the weight of j^{th} feature calculated using gain ratio measure and $\delta(x_j, x^i_j)$ is the distance between two values, which is 0 if $x_j = x_{ij}$ and 1 otherwise.

2.5 Neural Network Method

Neural network is one of the supervised method in which interconnection of artificial neurons are observed [6]. The Hidden Markow Model or back propagation based feed forward network are used in neural network method to disambiguate the ambiguous word. The inputs to learning techniques are pair of features and the expected outputs. This input features are used to partition the training contexts into non-overlapping sets according to required responses. The weights of neurons are adjusted according to the required outputs which are having larger values.

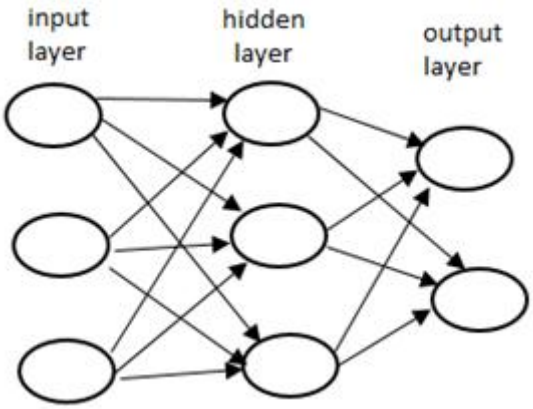

Figure 2: A feed forward neural network WSD with 3 features and 2 Responses

3. Semi–Supervised Approach

Semi-supervised approach uses both labeled and unlabeled data for training [9]. This method is so called "semi-supervised approach" because it uses data required for both supervised and unsupervised approaches. Most experimental studies shows that when unlabeled data is used with the combination of small quantity of labeled data then the machine learning efficiency will get increased and are giving better performance. Semi-supervised approach is also known as minimally supervised learning. Both supervised and semi-supervised methods are making assumptions of languages and its discourse for resolving the ambiguity.

In this method, we are given a set of m independent and distributed examples $(x_1,x_2,x_3.\ldots\ldots x_m)\in X$ with corresponding labels $(y1,y_2,y_3,\ldots\ldots y_m)\in Y$. In addition to this we also having n unlabeled examples $(x_{m+1}, x_{m+2}, \ldots..x_{m+n})\in X$. Semi-supervised learning gives higher accuracies with less effort on annotating data. The important semi-supervised algorithms are explained as follows.

3.1 Yarowsky Bootstrapping Method

One of the most successful uses of the bootstrapping approach in Natural Language Processing is made by the Yarowsky in 1995. The Yarowsky method is incremental and one of simple iterative algorithm which does not requires large training sets and depends only on relatively small number of instances of each sense. As semi-supervised method uses labeled instances, these labeled instances are then used as raw information to train the classifier initially using other supervised methods. The trained initial classifiers are then used to extract a larger training set from the remaining untagged corpus. The trained sets which are obtained above the particular threshold are kept for future to train the other untrained corpus for next iteration.

The training, retraining and re-labeling process is repeated until particular changes are not observed. The main feature of this approach is the ability to train more and more training sets from small amount of initial training data. The precision obtained is very high using this approach. The sets of training examples are increases as iteration proceeds. The recalls are found to be improved by such iterations.

3.2 Bilingual Bootstrapping Methods

A new method for word sense disambiguation, one that uses a machine learning technique called bilingual bootstrapping. In learning to disambiguate words to be translated, bilingual bootstrapping [10] makes use of a small amount of classified data and a large amount of unclassified data in both the source and the target languages. The data in the two languages should be from the same domain but are not required to be exactly in parallel. It repeatedly constructs classifiers in the two languages in parallel by repeating the following two steps: (1) Construct a classifier for each of the languages on the basis of classified data in both languages, and (2) use the constructed classifier for each language to classify unclassified data, which are then added to the classified data of the language. We can use classified data in both languages in step (1), because words in one language have translations in the other, and we can transform data from one language into the other.

It boosts the performance of the classifiers by classifying unclassified data in the two languages and by exchanging information regarding classified data between the two languages. The performance of bilingual bootstrapping been experimentally evaluated in word translation disambiguation, and all of their results indicate that bilingual bootstrapping consistently and significantly outperforms monolingual bootstrapping. The higher performance of bilingual bootstrapping can be attributed to its effective use of the asymmetric relationship between the ambiguous words in the two languages.

3.3 Label Propagation Algorithm

This algorithm works by representing labeled and unlabeled examples as vertices in a connected graph, then propagating the label information from any vertex to nearby vertices through weighted edges iteratively, finally inferring the labels of unlabeled examples after the propagation process converges. In LP algorithm [15] (Zhu and Ghahramani, 2002), label information of any vertex in a graph is propagated to nearby vertices through weighted edges until a global stable stage is achieved. Larger edge weights allow labels to travel through easier.

Thus closer examples, more likely they have similar labels (the global consistency assumption). In label propagation process, the soft label of each initial labeled example is clamped in each iteration to replenish label sources from

these labeled data. Thus the labeled data act like sources to push out labels through unlabeled data. With this push from labeled examples, the class boundaries will be pushed through edges with large weights and settle in gaps along edges with small weights. If the data structure fits the classification goal, then LP algorithm can use these unlabeled data to help learning classification plane.

4. Unsupervised Approach

Unsupervised approach for word sense disambiguation uses an un-annotated [11] raw data and it acquires knowledge by assuming some sort of similarity form clusters formed by the words. These approaches based on the idea that same sense of word will have similar neighboring words [12]. To disambiguate a word they using some measure of similarity in context to get the correct sense. Unsupervised WSD performs word sense discrimination i.e. it divides the occurrence of word into a number of classes by determining for any two occurrences whether they belong to the same sense or not. Evaluation of these methods is more difficult. Main task of unsupervised approaches are identifying sense clusters.

Unsupervised methods overcome the problem of knowledge acquisition bottleneck. The performance of unsupervised method is always been lower than that of the other method used for disambiguation. Different methods in unsupervised approach are word clustering method in which words are clustered according to the semantic similarity based on single feature (e.g., subject-verb, adjective-noun, etc.) i.e. they are similar to that of target word. In another context clustering method the clusters are formed by finding the co-occurrence of word (not target) with the target word and then the centroid is calculated of the vector of words occurring in the same context. In another method called graph based method in which a graph is built on some grammatical relationship, in graph weights are assign to the edge according to the relatedness. An iterative algorithm is applied to get the word with highest degree node and finally minimum spanning tree is applied to disambiguate instance of target word. Problems with Unsupervised Approach are the instances in training data may not be assigned the correct sense. Clusters are heterogeneous. Number of cluster may differ from the number of senses of target word to be disambiguated.

4.1 Context Group Discrimination

This algorithm, which is due to Schutze (1992), goes one step ahead to discriminate the word senses after their context vectors are formed. This algorithm was developed to cluster the senses of the words for which ambiguity is present in the corpus. The algorithm represents senses, words, and context in a multi-dimensional real-valued vector space. The clustering is done based on contextual similarities between the occurrences. The contextual similarities are still found with cosine function, but the clustering is done using Expectation Maximization algorithm, an iterative, probabilistic model for maximum likelihood estimation. In the sense acquisition phase, the contexts of all the occurrences of the ambiguous words are represented as context vectors as explained earlier, and a method called average agglomerative clustering is used. The similarity is calculated as a function of number of neighbors common to the words. The more similar words appear in the two contexts, more similar the contexts become. After this, the occurrences are grouped so that occurrences with similar contexts are assigned to same cluster. A very similar approach is followed in Structural Semantic Interconnections (hybrid algorithm).

4.2 Co-occurrence Graphs

Whereas the previous techniques use vectors to represent the words, the algorithms in this domain make use of graphs. Every word in the text becomes a vertex and syntactic relations become edges. The context units (*e.g.* paragraph) in which the target words occur, are used to create the graphs. The algorithm worth mentioning here is Hyperlex, as proposed by Veronis (2004).

4.2.1 Hyperlex

As per this algorithm, [16] the words in context (*e.g.* in the same paragraph) with the target word become vertices, and they are joined with an edge, if they co-occur in same paragraph. The edge weights are inversely proportional to the frequency of co-occurrence of these words.

4.3 WSD using parallel corpora

It was experimentally found out that, words in one language, which have multiple meanings, have distinct translations in some other language. This assumption is utilized by Ide et al. (2002) in an algorithm for disambiguation. The algorithm was designed with the aim of obtaining large sense marked corpus automatically annotated with high efficiency. For this purpose, the algorithm needs raw corpus from more than one language (hence the name parallel corpora).

5. Conclusion

We have studied the important approaches for word sense disambiguation namely, Supervised, Semi-supervised and Unsupervised learning approaches and their corresponding important methods which are used for disambiguation purpose. WSD is one important field while study of Natural language processing. WSD can be resolved using above approaches. Supervised approach is found to be less time consuming than semi-supervised and the unsupervised approaches because of type of data used.

References

[1] Eneko Agirre and Philip Edmonds, “Word Sense Disambiguation: Algorithms and Applications”, Springer, 2006.
[2] Fellbaum Christiane, “WordNet: An electronic Lexical database”, MIT Press, Map 1998.
[3] Leacock, Miller and Chodorow, “Using Corpus Statistics and WordNet Relations for Sense Identification. Computational Linguistics”, 24:1, 147–165, 1998.

[4] Navigli, Roberto, "Word Sense Disambiguation: A Survey", ACM Computing Surveys, 41(2), ACM Press, pp. 1-69, 2009.
[5] Yarowsky D., "Hierarchical Decision Lists for Word Sense Disambiguation", Computers and the Humanities, 34(2):179-186, 2000.
[6] Azzini, C. da Costa Pereira, Dragoni, Tettamanzi, "Evolving Neural Networks for Word Sense Disambiguation", WSPC – Proceedings, 2008.
[7] Alistair Kennedy and Stan Szpakowicz., "A Supervised Method of Feature Weighting for Measuring Semantic Relatedness", 2011.
[8] Sreedhar, Viswanadha Raju, Vinaya Babu, Amjan Shaik, Pavan Kumar, "Word Sense Disambiguation: An Empirical Survey", volume 2. IJSCE, 2012.
[9] Ankita Sati, "Review: Semi-Supervised Learning Methods for Word Sense Disambiguation", volume 12, issue 4. IOSR-JCE, 2013.
[10] Li, H. & Li, C., "Word Translation Disambiguation Using Bilingual bootstrapping", Computational Linguistics, 30(1), 1-22, 2004.
[11] Zhang & Kim, "Word Sense Disambiguation by Learning from Unlabeled Data", ACL-2000
[12] Yarowsky D., "Unsupervised Word Sense Disambiguation Rivaling Supervised Methods", ACL-1995, pp. 189-196, 1995.
[13] Gerard Escudero, Llu´ıs M`arquez and German Rigau, "Naïve Bayes and Exemplar-based approaches to Word Sense Disambiguation Revisited", 2000.
[14] David Martinez, Eneko Agirre, Xinglong Wang, "Word Relatives in Context for Word Sense Disambiguation", ALTW, pp 42-50, 2006.
[15] Zheng-Yu Niu, Dong-Hong Ji & Chew Lim Tan, "Word Sense Disambiguation Using Label Propagation Based Semi-Supervised Learning".
[16] Satanjeev Banerjee, Ted Pedersen, "An adaptive Lesk Algorithm for Word Sense Disambiguation Using WordNet", Proceedings of the Third International Conference on Computational Linguistics and Intelligent Text Processing, page no: 136-145, 2002.
[17] Karov, Y. & Edelman, S., "Similarity-Based Word Sense Disambiguation. Computational Linguistics", 24(1): 41-59, 1998.
[18] Brown P., Stephen, D.P., Vincent, D.P., & Robert, Mercer, "Word Sense Disambiguation Using Statistical Methods", ACL-1991.
[19] Schutze H., "Automatic Word Sense Discrimination Computational Linguistics", 24:1, 97–123, 1998.
[20] Ping Chen, Wei Ding, Max Choly, Chris Bowes, "Word Sense Disambiguation with Automatically Acquired Knowledge", volume 24. Intelligent System, IEEE, 2012.
[21] David Martinez, Eneko Agirre, Xinglong Wang, "Word Relatives in Context for Word Sense Disambiguation", ALTW, pp 42-50, 2006.