Unsupervised Word Sense Disambiguation

Survey

Shaikh Samiulla Zakirhussain
Roll No: 113050032

Under the guidance of
Prof. Pushpak Bhattacharyya

Department of Computer Science and Engineering
Indian Institute of Technology, Bombay

June 14, 2013

Contents

Chapter 1

Past work in unsupervised WSD

Before going to the work done in unsupervised WSD, let us first understand its importance. As we saw in the previous chapter, WSD is very tough problem and needs large number of lexical and knowledge resources like sense tagged corpora, machine readable dictionaries *etc*. It is evident that use of such resources improves the performance of WSD. Hence one might think that, if such resources are available, then why not use them? or why not spend sufficient time in creating high quality resources and perform great in terms of accuracy. The main reason is that, even if we have all possible resources to build a great supervised approach, it can not be ported to other language easily. The resources have to be replicated for all possible languages. Another disadvantage of using the supervised approaches is, by using fixed sense repositories, we constrain ourself to the fixed number of senses present in that repository. We can not discover new senses of words, which are not present in the sense repository. Hence only considering the accuracy of the approach is not a good idea, but considering its versatility and portability to other languages and domains is also equally important. This is the reason we see many unsupervised approaches being tried by many researchers in WSD.

One more important question is to determine, which approach should be really called as unsupervised. The term unsupervised WSD is itself ambiguous [Pedersen, 2006]. Generally, the approach which does not use any sense tagged corpora is termed as unsupervised. This definition includes approaches which may use manually created lexical resources other than sense tagged corpora, such as wordnet, multilingual dictionary *etc*. The other definition of unsupervised WSD can be, approaches which use only untagged corpora for disambiguation. These are mainly clustering approaches. They cluster words or contexts, and each cluster corresponds to a sense of a target word.

In the following part of the chapter, some good unsupervised WSD approaches have been described. Every approach has varying characteristics depending upon amount of resources used and the performance of the approach in different scenarios. Let us see them one by one.

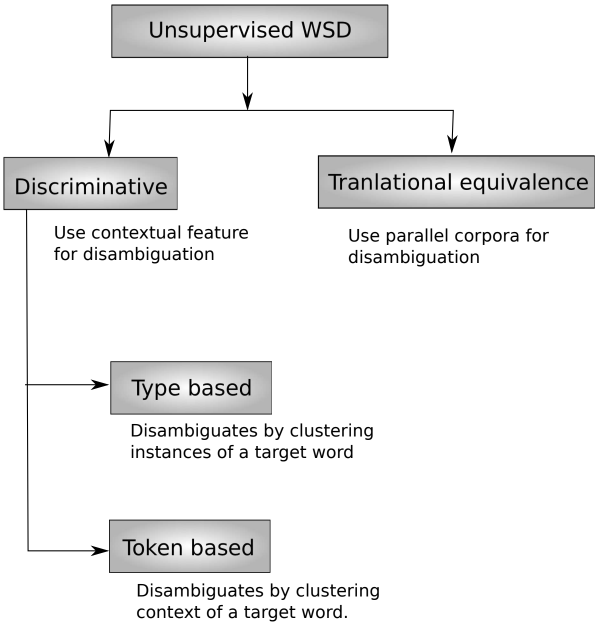
Unsupervised WSD
Discriminative
Use contextual feature for disambiguation
Tranlational equivalence
Use parallel corpora for disambiguation
Type based
Disambiguates by clustering instances of a target word
Token based
Disambiguates by clustering context of a target word.

Figure 1.1: Different approaches to Unsupervised WSD, [Pedersen, 2006]

1.1 Pedersen's approach of clustering the context

Ted Pedersen is one of the well known researchers in unsupervised WSD. He is known for his work in context clustering [Pedersen and Bruce, 1997]. Before understanding the actual approach, we will have a look at various types of unsupervised WSD approaches, which will help us understand the typical novelty of his approach. Unsupervised approaches are mainly of two kinds *viz.*, discriminative and translation based. Discriminative approaches are based on monolingual untagged corpora and discriminative context features while translation based approaches try to leverage parallel corpora for disambiguation. Discriminative approaches are classified as type-based and token-based. Type based approaches cluster various occurrences of the target words depending upon their contextual features while token based approaches cluster different contexts of a given target word. Various types of approaches are summarized in figure 1.1. Pedersen's approach is a token-based discriminative approach. The important feature of this approach is that it doesn't use any knowledge resource. He termed such approaches as *knowledge lean* approaches.

Pedersen proposed an unsupervised approach of context clustering [Pedersen and Bruce, 1997, Pedersen et al., 2005]. This is the target word WSD approach. The set of target words is selected initially. Each context of a target word is represented by a small feature vector which includes morphological features, the part of speech of surrounding words, and some co-occurrence features. A first order co-occurrence vector is created to represent each context. Co-occurrence features include co-occurrence vector corresponding to three most frequent words in the corpus, collocations with top twenty most frequent words and collocations with top twenty most frequent content words. Thus each cluster has been represented by a feature vector. All the contexts are represented by a $N \times M$ matrix. An $N \times N$ dissimilarity matrix is created in which each $(i, j)^{th}$ entry is the number of differing features in i^{th} and j^{th}

context. These contexts are clustered with McQuitty's average link clustering, which is a kind of agglomerative clustering algorithm. Every context is initially put into a separate cluster. Then most similar clusters are merged together successively. This process of merging clusters is continued until a specific number of clusters is reached or the minimum dissimilarity value among clusters crosses some threshold. Thus formed clusters are labeled in such a way that agreement with the gold data is maximized. The performance was compared among various clustering methods like Ward's agglomerative clustering and EM algorithm. Results show that McQuitty performed best among the three clustering methods.

1.2 HyperLex

This is a graph-based Unsupervised WSD approach proposed by [Veronis, 2004]. This is a target word WSD approach primarily developed for Information Retrieval applications. The approach was meant for identifying the paragraphs with the relevant sense of the target word. For a given target word, all nouns and adjectives in its context are identified, and represented as nodes in a co-occurrence graph. Verbs and adverbs were not considered because they reduced the performance significantly. Determiners and prepositions were removed. Even words related to web were removed as well *e.g.,* menu, home, link, http, *etc.* Words with less than 10 occurrences were removed and contexts with less than 4 words were eliminated. After all these filtering, finally, the co-occurrence graph for the target word is created. Only co-occurrences with frequency greater than 5 are considered. An edge is added between two vertices with weight defined as follows:

$$W_{A,B} = 1 - max[p(A|B), p(B|A)]$$

These probabilities are estimated by frequencies of A and B in corpus as follows:

$$p(A|B) = f(A,B)/f(B)$$

$$and$$

$$p(B|A) = f(A,B)/f(A)$$

Veronis stated that the graph thus created has the properties of "small worlds" [Watts and Strogatz, 1998]. "Small worlds" are characterized by the important phenomenon that any node in the graph is reachable from any other node in the graph within constant number of edges. *E.g.,* any individual on the planet is only ”six degrees away" from any other individual in the graph of social relations, even if there are several billion people. Another important characteristics of this kind of graphs is that there are many bundles of highly interconnected groups which are connected by sparse links. The highest degree node in each of these strongly connected components is known as root hub. Once the co-occurrence graph for the target word is constructed, the strongly connected components of the graphs are identified. Each strongly connected component is representative of the distinct sense of the target word. Root hubs are identified as the most connected nodes of each strongly connected component. Finding root hubs and the strongly connected components in a graph is an NP-hard problem. An approximate algorithm is used for this purpose whose approximation ratio is 2.

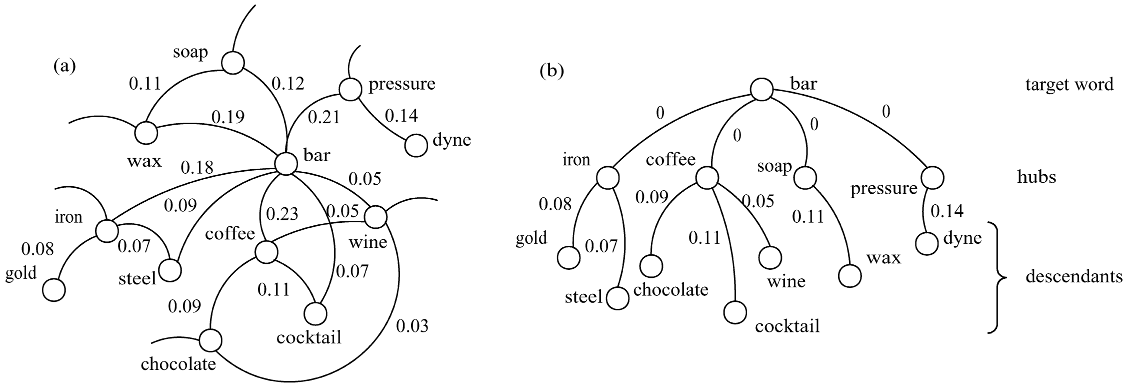

Figure 1.2: Hyperlex showing (a) Part of a cooccurrence graph. (b) The minimum spanning tree for the target word *bar*. (Figure courtesy [Navigli, 2009])

Once we have root hubs and strongly connected components, a node for the target word is then added to the graph. Target word is connected to each root hub with the zero edge weight, and the minimum spanning tree of the resulting graph is found. Now there exists a unique path from each node to the target word node (Note that each edge connected to target node will be present in the minimum spanning tree because of the zero edge weight). Each subtree is assigned a score which is the sum of the scores of the individual nodes in that subtree. The score of each sub-tree is found by following formula: Each node in the MST is assigned a score vector s with as many dimensions as there are components:

$$s = \begin{cases} \frac{1}{1+d(h_{i,v})} & if\, v \in component\, i \\ 0 & otherwise \end{cases}$$

where, $d(h_{i,v})$ is the distance between root hub h_i and node v in the tree.

The score vectors of all words are added for the given context. For the given occurrence of a target word, only the words from its context take part in the scoring process. The component with highest score becomes the winner sense.

The approach can be understood by the example in the figure 1.2. Figure 1.2 (a) shows the part of the co-occurrence graph for the word *bar*. Figure 1.2 (b) shows the minimum spanning tree formed after adding *bar* to the graph. Note that each subtree contains a set of words which represent a distinct sense of the target word *bar*.

Hyperlex was evaluated for 10 highly polysemous French words. It resulted in 97% precision. Note that this precision is for target word WSD that too restricted from nouns and adjectives. Performing good for verbs is difficult for an unsupervised algorithms.

1.3 PageRank

This is one more graph based approach to WSD, proposed by [Mihalcea et al., 2004]. It uses Wordnet as a sense inventory. It also uses semantic similarity measure based on Wordnet, which makes it knowledge based. But it does not use any sense tagged corpora for building a model, hence studying this approach under the title of unsupervised WSD is reasonable. But since there is a class of algorithms, which use only untagged corpora as a resource, we will term this approach as unsupervised knowledge-based approach.

The main idea of PageRank was proposed for ranking web-pages for a search engine. [Mihalcea et al., 2004] adapted this approach for application in WSD. PageRank is mainly used for deciding the importance of vertices in a given graph. The connection from node A to Node B represents that node A votes for node B. The score of every node is determined by the sum of the total incoming votes. The score of the vote is also proportional to the score of the incoming node. This process of voting is continued until the scores of the nodes converge to a stable value. After convergence, the score of every node represents its rank. The score of a vertex is defined as:

$$S(V_i) = (1-d) + d * \sum_{j \in In(V_i)} \frac{S(V_j)}{|Out(V_j)|}$$

Here, $(1-d)$ is the probability that user will jump randomly to current page. It is normally taken to be $0.85(d = 0.15)$. The ranks of the nodes are initialized arbitrarily in the beginning. This was about the actual PageRank algorithm. Now let us understand, how it was used as an unsupervised WSD algorithm.

All senses of all words are included in the graph, because every sense is a potential candidate for given words. Each node in the graph corresponds to a sense in the wordnet. Edges are taken from semantic relations in Wordnet. The senses sharing the same lexicalization are termed as competing senses, and no edges are drawn in between such senses. Some composite relations were also considered like sibling relation (Concepts sharing same hypernymy). Some preprocessing was done on the text before application of the PageRank algorithm. The text is tokenized and marked with part-of-speech tags. All the senses of the open class words except named entities and modal/auxiliary verbs were added to the graph. All the possible semantic relations between non-competing senses were added to the graph. After the graph is created, the PageRank is run on the graph with small initial value assigned to every node. After the algorithm converges, each node is assigned a rank. Each ambiguous word is tagged with a sense with the highest rank amongst its candidate synsets.

The algorithm was tested on SEMCOR and got 45.11% accuracy while Lesk algorithm got only 39.87% accuracy. PageRank was combined with Lesk and sense frequencies to get accuracy up to 70.32%.

1.4 Graph connectivity measures for unsupervised WSD

Navigli proposed a graph based Unsupervised WSD algorithm [Navigli and Lapata, 2007], in which a graph is constructed for every sentence using Wordnet, and graph connectivity measures are used to assign senses to the words in the sentence. For each sentence, a set of all possible senses of all words are determined using the sense inventory. Each sense becomes the node in the graph for that sentence. The set of nodes represent all possible meanings, the sentence can take. For every node in the graph, a DFS (Depth first Search) is initiated. If another node from the graph is encountered in between, all the intermediate nodes, along with the edges, are added to the graph. The depth first search is limited to six edges to reduce the complexity. Now, every node in the graph is at-most three edges away from nodes in the original sentence. Ranks are assigned to the vertices in the order of their local graph connectivity measures. Local graph connectivity measures help in determining the sense of the individual word, while global graph connectivity measures help in determining the overall meaning of the sentence. Using assigned ranks, the meaning of the sentence corresponding to the maximum global graph connectivity is assigned to the sentence. Intuition behind this approach is simple. The sense combination is most probable if the chosen senses are most strongly connected to each other.

WordNet 2.0 and the extended WordNet, which contains additional cross part-of-speech relations, were used as sense inventories. Various local connectivity measures *viz.*, In-degree Centrality, Eigenvector Centrality, Key Player Problem, Betweenness Centrality, Maximum flow were used as local graph-connectivity measures. Compactness, Graph Entropy and Edge Density were used as global graph connectivity measures. It was seen that Key Player Problem (KPP) measure performed best among local connectivity measures while Compactness performed best amongst global similarity measures. Local measures performed significantly better than global measures, while the performance of the algorithm increases with increase in number of edges considered.

1.5 Disambiguation by Translation

Disambiguation by translation is very interesting approach under unsupervised WSD. All the approaches we saw by now use the untagged corpus of only one language with some knowledge resources. As opposed to that, disambiguation by translation uses untagged word-aligned parallel corpora in two languages. Translations are very strong clue for disambiguation. Looking at the translation of a given polysemous word, we can restrict the number of possible senses to the intersection of senses of the target word and its translation. For using parallel text, we have to first align it. Sentence alignment and word alignment of parallel corpora can be done either manually or using GIZA++ [Och and Ney, 2000]. Once the alignment is done, we can use the translations of the target word to disambiguate it. Some good approaches of this kind are [Ide et al., 2002], [Gale et al., 1992], [Diab and Resnik, 2002] and [Ng et al., 2003]. We will have a look at the approaches by [Ide et al., 2002] and [Diab and Resnik, 2002].

1.5.1 Sense Discrimination with parallel corpora

Defining the sense granularities is a difficult task for WSD. Working with predefined sense inventories imposes restrictions on WSD by not allowing the discovery of new senses, and by unnecessarily considering too fine grained senses which may not be necessary for the target domain. [Ide et al., 2002] came up with a parallel corpora based approach for defining the sense discriminations and using them for performing WSD. They defined the senses of the words through their lexicalizations in other languages. They claim that sense discrimination obtained by their algorithm is at least as good as that obtained by human annotators. Thus obtained sense discriminations can suit best for various NLP applications like WSD.

They took the parallel corpora in 6 languages and defined sense discriminations using the translation correspondences. Initially, every translation is assumed to be a possible sense of a target word. Then all these senses are clustered using an agglomerative clustering algorithm. The resulting clusters are taken to represent senses and the sub-senses of the target word. Senses thus obtained were normalized by merging the clusters which are very close and flattening the hierarchical senses to match the flat wordnet representation. These flat senses were then matched with the senses assigned by the human annotators. The agreement between clusters and annotators was comparable to that between two annotators. These discriminations are used to sense tag the corpora with appropriate senses. They showed through their results that coarse grained agreement is the best that can be expected from humans, and that their method is capable of duplicating sense differentiation at this level.

1.5.2 Unsupervised WSD using parallel corpora

This approach [Diab and Resnik, 2002] exploits the translation correspondences in parallel corpora. It uses the fact that the lexicalizations of the same concept in two different languages preserve some core semantic features. These features can be exploited for disambiguation of the either lexicalizations. The approach sense tags the text in the source language using the parallel text and the sense inventory in the target language. In this process, the target language corpus is also sense tagged. In the experiments performed by the author, French was the source language and English was the target language. English-French parallel corpus and the English sense inventory was used for experimentation.

The algorithm is divided into four main steps:

- In the first step, words in the target corpus (English) and their corresponding translations in the source corpus (French) are identified.
- In the second step, target sets are formed by grouping the words in the target language.
- In the third step, within each of these target sets, all the possible sense-tags for each word are considered and then sense-tags are selected which are informed by semantic similarity with the other words in the group.
- Finally, sense-tags of words in target language are projected to the corresponding words in the source language. As a result, a large number of French words received tags from English sense inventory. As a result, a large number of French words received tags from English sense inventory.

Let us understand this process with example of Marathi as a source language and Hindi as a target language. Parallel aligned untagged texts in Hindi and Marathi and the

Hindi sense inventory will be used for disambiguation. Note that this illustration is just for the sake of understanding, no actual experimentation was done in Hindi and Marathi languages by us.

- Suppose an occurrence of the Marathi word फळ is aligned with the Hindi word फल.
- Then we will find the target set of the word फळ, which will be something like {फल, परिणाम, परिणती}.
- Now we will consider all the senses of all words in the target set *viz.*, 662, 4314 and 2035. Looking at the words in the target set gives an idea about the sense of the target word. Most probable sense inferred by the target set is 2035. The sense which gives maximum semantic similarity among the words in target set is the winner sense. The similarity measure by [Resnik and Yarowsky, 1999].
- Finally, sense-tags of words in target language (2035 in this case) are projected to the corresponding words in the source language.

Performance of this approach has been evaluated using the standard SENSEVAL-2 test data and results showed that it is comparable with other unsupervised WSD systems.

1.6 WSD using Roget's Thesaurus categories

Roget's thesaurus is an early Nineteenth century thesaurus which provides classification or categories which are approximations of conceptual classes. This algorithm by [Yarowsky, 1992] uses precisely this ability of Roget's thesaurus to discriminate between the senses using statistical models. The algorithm observes following:

- Different conceptual classes of words tend to appear in recognizably different contexts.
- Different word senses belong to different conceptual classes.
- A context based discriminator for the conceptual classes can serve as a context based discriminator for the members of those classes.

The algorithm thus identifies salient words in the collective context of the thesaurus category and weighs them appropriately. It then predicts the appropriate category for an ambiguous word using the weights of words in its context. The prediction is done using:

$$\underset{\text{RCat}}{\text{argmax}} \sum_{w \in context} log\,(\frac{Pr(w|RCat)*Pr(RCat)}{Pr(w)})$$

where, *RCat* is the Roger's thesaurus category.

The following table shows the implementation of Yarowsky's algorithm on the target word *crane*. A *crane* might mean a machine operated for construction purpose (Roget's category of TOOLS/MACHINE) or a bird (Roget's category of ANIMAL/INSECT). By finding the context words for word *crane* and finding how much weight (similarity) they impose on each sense of *crane*, the winner sense is selected.

TOOLS/MACHINE	*Weight*	ANIMAL/INSECT	*Weight*
lift	2.44	Water	0.76
grain	1.68		
used	1.32		
heavy	1.28		
Treadmills	1.16		
attached	0.58		
grind	0.29		
Water	0.11		
TOTAL	11.30	*TOTAL*	0.76

Table 1.1: Example list showing a run of Yarowsky's algorithm for the senses of the word *crane* belonging to (a) TOOLS/MACHINE and (b) ANIMAL/INSECT domains along with weights of context words. The highlighted sense is the winner sense.

Bibliography

[Diab and Resnik, 2002] Diab, M. and Resnik, P. (2002). An unsupervised method for word sense tagging using parallel corpora. In Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, ACL '02, pages 255–262, Morristown, NJ, USA. Association for Computational Linguistics.

[Gale et al., 1992] Gale, W., Church, K., and Yarowsky, D. (1992). Estimating upper and lower bounds on the performance of word-sense disambiguation programs. In Proceedings of the 30th Annual Meeting of the Association for Computational Linguistics.

[Ide et al., 2002] Ide, N., Erjavec, T., and Tufis, D. (2002). Sense discrimination with parallel corpora. In Proceedings of the ACL-02 workshop on Word sense disambiguation, pages 61–66, Morristown, NJ, USA. Association for Computational Linguistics.

[Mihalcea et al., 2004] Mihalcea, R., Tarau, P., and Figa, E. (2004). Pagerank on semantic networks, with application to word sense disambiguation. In Proceedings of Coling 2004, pages 1126–1132, Geneva, Switzerland. COLING.

[Navigli, 2009] Navigli, R. (February 2009). Word sense disambiguation: A survey. In ACM Computing Surveys, Vol. 41, No. 2, Article 10.

[Navigli and Lapata, 2007] Navigli, R. and Lapata, M. (2007). Graph connectivity measures for unsupervised word sense disambiguation. In Veloso, M. M., editor, IJCAI, pages 1683–1688.

[Ng et al., 2003] Ng, H. T., Wang, B., and Chan, Y. S. (2003). Exploiting parallel texts for word sense disambiguation: an empirical study. In Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1, ACL '03, pages 455–462, Morristown, NJ, USA. Association for Computational Linguistics.

[Och and Ney, 2000] Och, F. J. and Ney, H. (2000). Improved statistical alignment models. In ACL00, pages 440–447, Hongkong, China.

[Pedersen, 2006] Pedersen, T. (2006). Unsupervised corpus-based methods for wsd. In Agirre, E. and Edmonds, P., editors, Word Sense Disambiguation, volume 33 of Text, Speech and Language Technology, pages 133–166. Springer Netherlands.

[Pedersen and Bruce, 1997] Pedersen, T. and Bruce, R. F. (1997). Distinguishing word senses in untagged text. CoRR, cmp-lg/9706008.

[Pedersen et al., 2005] Pedersen, T., Purandare, A., and Kulkarni, A. (2005). Name discrimination by clustering similar contexts. In Gelbukh, A. F., editor, CICLing, volume 3406 of Lecture Notes in Computer Science, pages 226–237. Springer.

[Resnik and Yarowsky, 1999] Resnik, P. and Yarowsky, D. (1999). Distinguishing systems and distinguishing senses: new evaluation methods for word sense disambiguation. Nat. Lang. Eng., 5:113–133.

[Veronis, 2004] Veronis, J. (2004). Hyperlex: Lexical cartography for information retrieval. Comput. Speech Lang., 18(3).

[Watts and Strogatz, 1998] Watts, D. J. and Strogatz, S. H. (1998). Collective dynamics of'small-world'networks. Nature, 393(6684):409–10.

[Yarowsky, 1992] Yarowsky, D. (1992). Word-sense disambiguation using statistical models of roget's categories trained on large corpora. In Proceedings of the 14th conference on Computational linguistics, pages 454–460, Morristown, NJ, USA. Association for Computational Linguistics.