Literature Survey on Unsupervised Word Sense Disambiguation

Devendra Singh Chaplot
Roll No: 100050033

Under the guidance of
Prof. Pushpak Bhattacharyya

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

May 7, 2014

Acknowledgement

I thank my guide Prof. Dr. Pushpak Bhattacharyya for his constant encouragement for rigorous study, and his invaluable guidance. I thank my colleague, Sudha Bhingardive for the minutely detailed discussions we had on algorithms from the literature survey and their analysis. I thank Salil Joshi, M.tech. student, now working with IBM Research, for assisiting me throughout the process of research and and helping me time to time. I also thank all my WSD group members Diptesh Kanojia, Siddartha Gunti, Brijesh Bhatt and Subhash Kunnath. I also thank my batch mates Priyank Chhipa and Ashwin Paranjape for their help and support. This report could not have been complete without any of them.

Chapter 1

Introduction

Word sense disambiguation (WSD) is the ability to identify the meaning of words in context in a computational manner. WSD is considered an AI-complete problem, that is, a task whose solution is at least as hard as the most difficult problems in artificial intelligence.

-Roberto Navigli [Navigli, 2009]

Natural languages are ambiguous. A word can take several senses depending on the context in which it appears. Let's consider following two sentences:

That is an interesting **problem** to solve.
There is some **problem** in this gadget.

In the first sentence, the word '*problem*' means 'a question raised for consideration or solution'. But in the second sentence, it means 'a source of difficulty'. Thus, the same word can take different meanings depending on the context. Identifying the correct sense of polysemous words seems very easy at first, but its not so trivial, if we think computationally. It involves processing unstructured information to extract the underlying meaning.

1.1 Homonymy and polysemy

Ambiguity is inherent in natural languages. Different aspects of ambiguity are categorized based on their nature. In context of WSD, a couple of sense related ambiguities *viz., Homonymy* and *polysemy*, are worth mentioning here.

1. **Homonymy:** A homonym is a group of words sharing same spelling and pronunciation but have different senses. These multiple senses can be totally unrelated to each other. *E.g.,* the word '*left*' can mean '*past form of the verb leave*' or *opposite of right*. Homonyms are the targets of WSD.

2. **Polysemy:** Polysemy is the special case of homonymy where multiple senses of the word are related to each other. These senses share the same root. *E.g.,* the word

mouth means 'mouth of the animal' as well as 'the mouth of the river or the cave'. For WSD, polysemy is more difficult to handle as compared to other homonyms, since it is difficult to differentiate between closely related senses.

1.2 Problem description

WSD is the ability to find the appropriate sense of the target word using the contextual information. WSD is performed on a piece of a given text, which is a set words. It can be thought of as a classification task where each word can be classified into one of its possible candidate senses. There are two major variations of WSD:

1. **Lexical Sampling (Target word WSD):** A restricted set of target words is taken. This task focuses on disambiguation of this restricted set of words. Supervised systems are generally used for this task because system can be trained for each of the target word using manually tagged data.

2. **All word WSD:** All word WSD expects the disambiguation of all the content words in the given corpus. Here, supervised systems may face the problem of data sparsity. Hence, wide coverage knowledge based or unsupervised approaches are used for this task.

1.2.1 Selecting the senses

Before disambiguation, knowing the possible senses of each of the target word is important. These candidate synsets can either be acquired from the sense inventories or can be discovered ad-hoc during the disambiguation process. Using the sense inventories usually help but may create problems when either the expected senses are not covered by the sense repository or it has too many fine grained senses unnecessarily confusing the algorithm. Selection of senses usually depends on the application. Mostly, the listed senses from the sense repositories like wordnets, thesauri, machine readable dictionaries, ontologies *etc.* are used.

1.2.2 Representation of context

Context gives the clue about the sense of the target word. But it is an unstructured piece of text which should be represented in a proper usable format which preserves the necessary information. Depending on the requirement of the algorithm, context can be represented as the combination of various features like lemma, POS-tag, morphology, semantic relations *etc*. For obtaining these features from the text, we need some preprocessing on the raw text which includes morphology analysis, POS tagging, chunking, parsing *etc*.

1.3 Choosing the approach

Broadly speaking, statistical WSD approaches are classified in two major categories depending on the corpora they use. Supervised approaches need sense tagged corpora for training the model where as Unsupervised approaches work on untagged corpora. There are some semi-supervised approaches which work with small annotated seed data. Another way to classify WSD approaches is based on the use of knowledge resources. Approaches using knowledge resources like wordnet, ontologies, thesauri are termed as knowledge based approaches, while there are some approaches which work without any of these resources, which are termed as knowledge-lean or corpus based approaches. Let's have a look at these categories in brief.

1.3.1 Supervised WSD

Approaches relying on sense tagged corpora for disambiguation are known as supervised WSD approaches. They yield very high accuracy in the domain of the training corpus. But this accuracy comes at the cost of sense tagged corpora which is a costly resource in terms of the time and the manual efforts involved. Creating such corpora for all languages in all domains will be impracticable. Hence these approaches cannot be easily ported to different languages or domains. Some good supervised approaches are mentioned below.

1. **Decision List:** Decision lists[Rivest, 1987] can be viewed as a list of weighted if-then-else rules sorted in the descending order of weights. The rules are learned from the tagged corpus. These rules are of the form (features, sense, weight), where feature is the feature vector of the target word. The weight shows the classification potential of the rule. The rule with the higher weight is applied first. For any word w and its feature vector, the decision list is checked, and the sense with the highest score is assigned to w as follows.

   $$\hat{S} = argmax_{S_{i \in Senses_D(w)}} Score(S_i)$$

   The score of the sense S_i is calculated as the maximum score over features, which is defined as follows:

   $$Score(S_i) = max_f \; log\left(\frac{P(S_i|f)}{\sum_{i \neq j} P(S_j|f)}\right)$$

   Decision list was the most successful approach in the first Senseval competition. Table 1.1shows the simplified sample of a decision list for disambiguating the word मान in Marathi.

2. **Support Vector Machines:** Support Vector Machines [Boser et al., 1992] are based on the idea of learning a hyper-plane, from a set of the training data. The hyper-plane separates positive and negative examples. The hyper-plane is located in the feature space, such that it maximizes the distance between the closest positive and negative examples (called support vectors). SVM thus minimizes the classification error and maximizes the geometric distance or margin between the positive

Feature	Prediction	Score
मान with faminine gender marker	मान / part of body	4.83
मान with masculine gender marker	मान / respect	3.35
मान with दुखणे / V	मान / part of bidy	2.48
मान with राखणे / V	मान / respect	2.33
मान with आहे	————	0.01

Table 1.1: Example of a decision List for the Marathi word मान

and negative examples. The linear SVM is characterized by two parameters: w, which is the vector perpendicular to the hyper-plane and b, the bias which is offset of the hyper-plane from the origin.

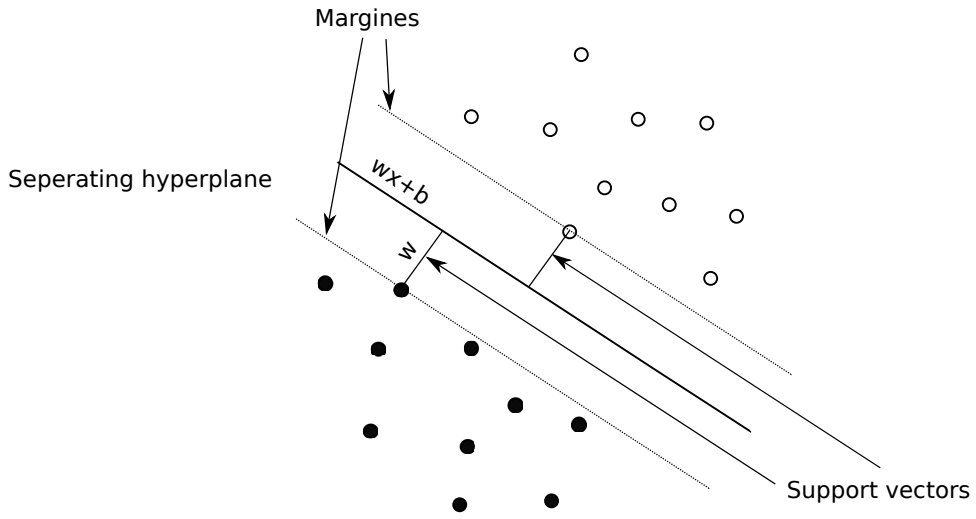

Figure 1.1: Support vector machine (geometric intuition)

An instance is labeled as positive if the value $f(x) = wx + b \geq 0$ and negative otherwise. Figure 1.1 shows the support vectors and the separating hyper-plane along with w and b. This can thus be well understood from the geometric intuition given in the figure. SVM is a binary classifier, but WSD is a multi-class problem, as there can be more than two senses (classes) for a word. To make it usable for WSD, the problem can be broken down into a number of binary class problems. This can be done by taking each sense as one class and the remaining senses as another class. This is done for all the senses. The sense with the maximum confidence score is taken as the winner sense. The confidence score is actually the value of $f(x) = wx + b$, for each SVM.

1.3.2 Semi-supervised WSD

Supervised algorithms are restricted to few languages because of their dependence on sense tagged corpora. Semi-supervised, also known as *minimally* supervised algorithms make some assumptions about the language and discourse in order to minimize these restrictions. The common thread of operation of these algorithms are these `assumptions` and the `seeds` used by them for disambiguation purposes.

This subsection presents two such approaches, based on two different ways to look at the problem, namely Bootstrapping and Monosemous Relatives.

1. **Bootstrapping:** This algorithm, devised by [Yarowsky, 1992], is based on Yarowsky's supervised algorithm that uses Decision Lists. As mentioned earlier, the algorithm makes a couple of assumptions regarding the language. The assumptions can be stated as follows:

   - **One sense per Collocation** - The sense of a word is strongly dependent on the neighboring words.
   - **One sense per Discourse** - Every document contains a single sense of a word with high probability.

   It can be seen that these assumptions are very strong, and thus the model building phase becomes quite small compared to the supervised analogue of this algorithm. With these assumptions, the algorithm first identifies a set of seed words, which can act as disambiguating words. A Decision List is built based on this seed data. Next, the entire sample set is classified using the Decision list generated previously.

   Using this decision list, as many new words as possible are classified in order to identify their senses. Using these words along with their identified senses, new seed data is generated. The same steps are repeated until the output converges up to a threshold value. Figure 1.2 shows the snapshots of the working bootstrapping for two senses of the word plant *viz.,* the living plant and the manufacturing plant.

2. **Monosemous Relatives:** With exponential growth of the *world wide web*, approaches are being tried out which can use the vast collection of words as corpus. This enables the algorithms to have an automatically annotated corpus, which has tremendously huge size, the *web corpus*.

   Monosemous relatives approach is developed as a bootstrapping algorithm to use words with single sense as possible synonyms. For this, through the synset of a word *w*, all words having single sense (the sense of *w* itself) are found. For each word s $\in$ this set, a web search is done and contexts are found. These contexts are directly sense annotated with sense of word *w*. A small variant here is to create *topic signatures* containing closely related words associated with each word sense. A manual inspection is necessary for such approaches.

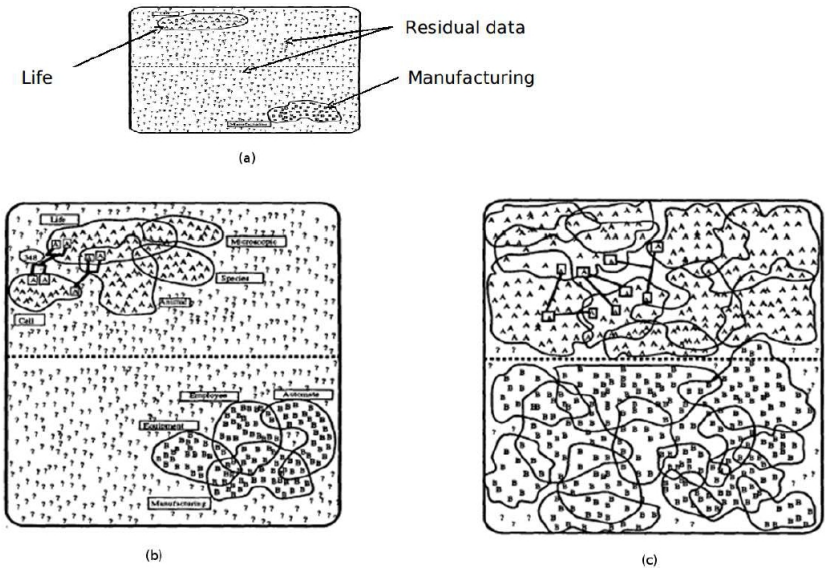

Figure 1.2: figure showing growth of Semi-supervised decision list on two senses of plant *viz.*, life and manufacturing. (a) The initial seed data. (b) Growth of the seed set. (c) Seed data converges. (Figure courtesy [Yarowsky, 1992])

1.3.3 Unsupervised WSD

Looking at the costly resources needed and the restricted nature of supervised algorithms, unsupervised WSD has become quite popular nowadays. Despite of their less accuracy, unsupervised WSD is chosen over supervised one many times because of its resource consciousness and robustness. Unsupervised approaches can be easily ported to other languages due to minimal dependencies. Unsupervised WSD is discussed in detail in chapter 2.

1.3.4 Knowledge based WSD

This is an entirely different category of approaches as compared to above mentioned statistical categories. It uses various knowledge resources like wordnets, thesori, ontologies *etc*. Their performance is poorer as compared to statistical approaches but they are known for their wide coverage. Lesk's approach [Lesk, 1986] based on overlap of sense definitions, is a good example of knowledge based WSD.

- **Lesk's approach:** In this approaches, the sense of the target word is determined base on the overlap among its context and the sense definitions from the machine readable dictionaries. The sense whose gloss has maximum number of words in common with the context is assigned to the target word. Each sense of the target word w gets the score as follows:

  $$score(S) = |context(w) \cap gloss(S)|$$

  This approach is very sensitive to the exact wording in the sense definitions and hence performed poorly. An improvement in this approach was suggested by

[Banerjee and Pedersen, 2002]. They extended the gloss of the sense by the glosses of its related synsets. The semantic relations in the wordnet like hypernymy, hyponymy, meronymy *etc*. are used for expanding the gloss. This improvement resulted in much better accuracy (34.6% as compared to 18.3% for Lesk's algorithm). Still it is not comparable to the performance of statistical approaches.

1.4 WSD, an AI-Complete problem

Polysemy is an important feature of natural languages. Despite of this ambiguity, humans can very well communicate through this kind of languages. They can easily disambiguate polysemous words using domain knowledge, context and discourse. This job of disambiguation among different senses of words is, however, fairly difficult for machines. Taking the flexible structure of human languages into account, making machines do this job is very difficult. Before they can perform such task, they should be equipped with the good quality knowledge resources like sense inventory, semantics *etc*. Even after providing such resources in machine readable format, the problem doesn't become simple. The versatile nature of human languages makes it difficult to capture every facet of their structure and semantics in a deterministic form. Hence even after using all available lexical and knowledge resources, WSD has been a challenging job for machines.

Polysemy is not the only challenge WSD has to face. If we have a closer look at natural languages, we find many beautiful syntactic and semantic constructs like idioms, metaphors, euphemism, pun, irony *etc*. *E.g.,* let us consider the idiom usage *"This task is a piece of cake for me!"*, where the idiom *"piece of cake"* actually means 'a fairly trivial task'. Here it is difficult to detect that the group of words actually means something else than the usual word to word meaning. Consider another example , "He became a mathematics teacher due to some *prime factors*". This is a good example of pun, where the word '*prime*' have two meanings viz. 'indivisible' and 'important'. Usually WSD algorithms find the most probable sense of the target word. Detecting such tricky uses of the words, where more than one senses of the words are depicted simultaneously for ornamental usage is a hard task.

WSD has been termed as an AI-complete problem by Navigli [Navigli, 2009]. With the analogy to the definition of NP-complete, AI-complete means that the problem is as hard as any other problem in AI. NLP is the subset of AI and WSD belongs to NLP hence WSD is NLP-complete as well. The problem is termed hardest in AI because of many factors. One being the representation of word senses. Senses can be represented at many levels of granularity. The main issue is to decide the refinement level to which the sense discrimination should be considered. Other important reason behind the complexity of the problem is heavy dependence on knowledge. Without knowledge, it will be impossible to disambiguate for machines and even for humans. But the need of knowledge varies from text to text, domain to domain. Availability of knowledge resources with equal refinements for all domains and languages is another issue adding to the complexity of the WSD task.

1.5 Importance of WSD in NLP

WSD is one of the most fundamental tasks in NLP. Many applications in NLP directly or indirectly rely on WSD. Sentiment Analysis, Machine Translation, Information Retrieval, Text summarization, Text Entailment, Semantic Role Labeling are some of the main applications in NLP which depend on WSD. WSD can be thought of as a heart of NLP. Significance of WSD in NLP can be understood by figure 1.3.

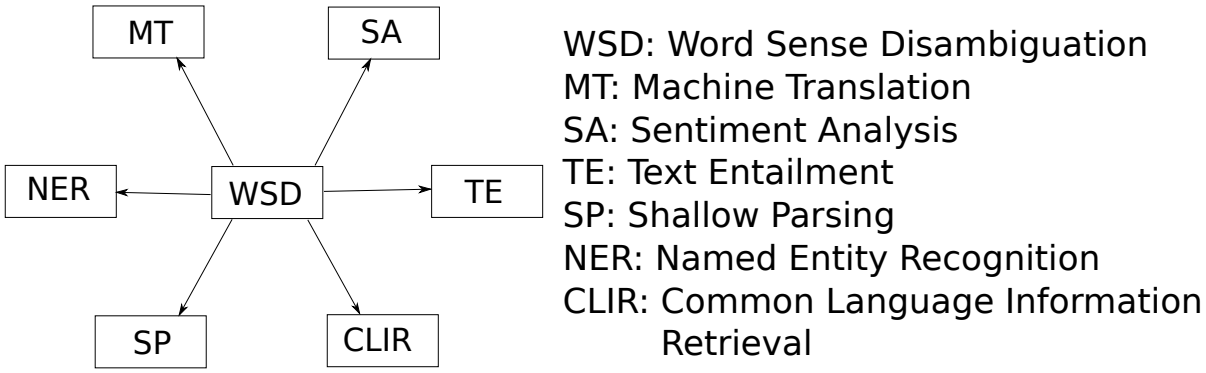

Figure 1.3: WSD as a heart of NLP

Many researchers have shown that WSD can improve the performance of SMT [Carpuat and Wu, 2007, Chan and Ng, 2007, Vickrey et al., 2005]. Let us consider a simple example. Suppose that following sentence is to be translated to Hindi.

"The table shows the statistics."

The word '*table*' has two major senses *viz.,* the furniture and the set of data arranged in rows and columns. So, we don't know which sense of the word '*table*' is expected in the sentence without looking at the context and the domain. In case we just translate every word in a random sense, the meaning of the sentence can change drastically. The translation can become something like 'यह मेज आँकडे दिखाता है।'. Conversely, if we first tag this sentence using some WSD algorithm as follows:

"The table_4386330 shows_2153218 the statistics_06030848."

Since the sentence is tagged with appropriate senses, the process of translation will become fairly easy. Now the translation will be as expected:

"यह तालिका आँकडे दिखाती है।"

Information Retrieval is another very important field where WSD is of great importance. *E.g.,* suppose a user is searching for the *'plane crash incident'*. Here the word *plane* is ambiguous. The first meaning is the 'air-plane' and the other being 'flat surface'. The query may retrieve several documents with the second sense of *plane* along with the relevant documents. If every document is indexed with the sense-tag instead of the word, the search will be very efficient. Hyperlex [Veronis, 2004] is a very good example where WSD is successfully used for IR.

Another example can be taken from Text entailment. Consider the following sentence:

"I dreamt a fairy scene yesterday.".

Here, the verb *dream* has two possible senses. One sense is 'a day-dream' and the other is 'the feel of reality during sleep'. Here the second sense entails sleeping, while the first sense does not entail anything about sleeping. After the sentences is sense tagged, the job of entailment will become very easy.

1.6 Organization of the report

The remainder of this report is organized as follows. Chapter 2 covers the past work done in Unsupervised WSD. Chapter **??** introduces the datasets and the knowledge resources used for experimentation. Chapter **??** describes IndoWordnet which is the rich multilingual semantic network of Indian languages. Chapter **??** covers the Expectation Maximization based algorithm for Unsupervised WSD by [Khapra et al., 2011]. Another approach incorporating contextual information in this EM based algorithm is described in Chapter **??**. Chapter **??** describes the recently published IndoWordnet Visualizer, which is graphical user interface to browse and explore the IndoWordnet lexical database for various Indian languages and how it helps in error analysis of our WSD system. This is followed by Error Analysis in Chapter **??**. Chapter **??** concludes the report by summarizing the aims and the achievements of the work and enlists possible future work, which can be expected ahead of this work.

Chapter 2

Survey on Unsupervised Word Sense Disambiguation

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.

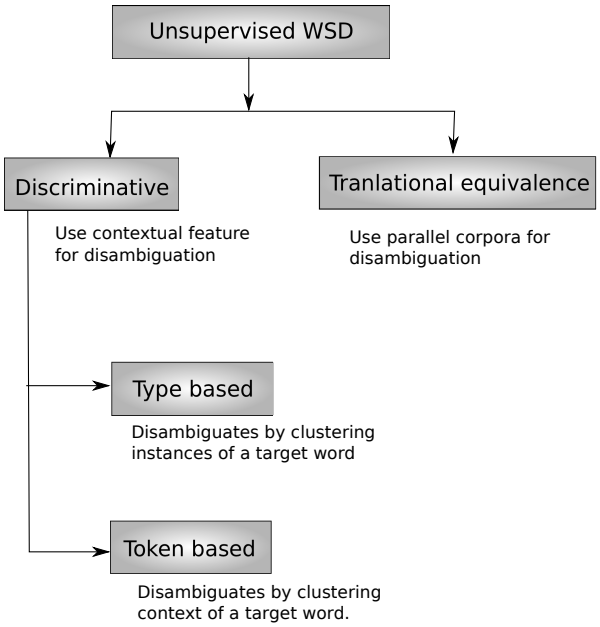

Figure 2.1: Different approaches to Unsupervised WSD, [Pedersen, 2006]

2.1 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 2.1. 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.

2.2 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.

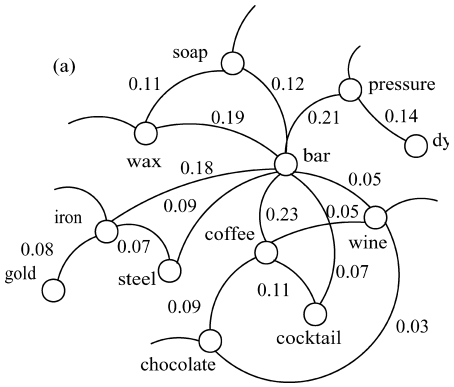

Figure 2.2: Hyperlex showing (a) Part of a cooccurrence graph. (b) The minimum spanning tree for the target word *bar*. (Figure courtesy [Navigli, 2009])

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 2.2. Figure 2.2 (a) shows the part of the co-occurrence graph for the word *bar*. Figure 2.2 (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.

2.3 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%.

2.4 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.

2.5 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].

2.5.1 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.

2.5.2 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.

2.6 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:

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

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 2.1: 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.

2.7 Summary

In this chapter, we have discussed the need for unsupervised WSD and its advantages over supervised techniques. We have also described various types of techniques for unsupervised WSD. The techniques which use contextual features for disambiguation are known as Discriminative while techniques based on Translation equivalence use parallel corpora for disambiguation. We have discussed Pederson's approach of clustering the context, Hyperlex and Page Rank algorithm, graph connectivity measures, disambiguation by translation and WSD using Roget's Thesaurus categories.

Bibliography

[Banerjee and Pedersen, 2002] Banerjee, S. and Pedersen, T. (2002). An adapted lesk algorithm for word sense disambiguation using wordnet. In CICLing '02: Proceedings of the Third International Conference on Computational Linguistics and Intelligent Text Processing, pages 136–145, London, UK. Springer-Verlag.

[Boser et al., 1992] Boser, B. E., Guyon, I. M., and Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory, pages 144–152. ACM Press.

[Carpuat and Wu, 2007] Carpuat, M. and Wu, D. (2007). Improving statistical machine translation using word sense disambiguation. In In Proc. of EMNLP-CoNLL.

[Chan and Ng, 2007] Chan, Y. S. and Ng, H. T. (2007). Domain adaptation with active learning for word sense disambiguation. In Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, pages 49–56, Prague, Czech Republic. Association for Computational Linguistics.

[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.

[Khapra et al., 2011] Khapra, M., Joshi, S., and Bhattacharyya, P. (2011). Help me and i will help you: A bilingual unsupervised approach for estimating sense distributions using expectation maximization. In 5th International Conference on Natural Language Processing (IJCNLP 2011), Chiang Mai, Thailand, November 2011.

[Lesk, 1986] Lesk, M. (1986). Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In In Proceedings of the 5th annual international conference on Systems documentation.

[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.

[Rivest, 1987] Rivest, R. L. (1987). Learning decision lists. Machine learning, 2(3):229–246.

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

[Vickrey et al., 2005] Vickrey, D., Biewald, L., Teyssier, M., and Koller, D. (2005). Word-sense disambiguation for machine translation. In In EMNLP, pages 771–778.

[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.