CHAPTER 3

LITERATURE REVIEW

In this chapter, we will firstly proceed with a discussion of the problem we are trying to solve, the definition of word senses in the context of Word Sense Disambiguation (WSD). This introduction into WSD will be followed by an overview of previous work done in the field of WSD, attempting a division according to the approach used. By approach or strategy, we refer to the primary resource used to extract information about the different senses of words.

3.1 What is a Word Sense?

The task of a WSD system is to resolve the lexical ambiguity of a word in a given context. The term lexical ambiguity refers to two different concepts: homonymy and polysemy. *Homonymy* describes the fact that two words have the same lexical form, but different etymologies and unrelated meanings (e.g. bank "financial institution" versus "river bank") whereas *polysemy* refers to one word having several related meanings (e.g. line meaning `thread', `row', `course of conduct', etc.). The meaning of "head", which can mean the upper or top part of our body or the top position, is another example of polysemy word.

The process of WSD is that WSD systems first preprocess the input sentence containing ambiguous words to extract a set of features used for disambiguation. This preprocessing typically involves morphological or syntactic analysis relations because the part-of-speech of words appearing in the input and syntactic relation involving ambiguous words can be informative features. Thereafter, by using the extracted features, the WSD system determine what the possible senses are and which sense is being used in the given instance.

The first step involved in the task of WSD is the determination of different senses for all words in the text to be disambiguated. The determination of senses can either be exhaustive, i.e. all possible meanings of a given word are identified, or tuned to the particular domain of the text under consideration. Most recent work in WSD relies on predefined senses consisting of either dictionary senses, a group of features or categories (as in a thesaurus), or translations from other languages.

One of the most difficult issues in applied lexical semantics is the definition of word senses. In dictionaries, each word is listed with a number of discrete senses and subsenses, possibly different from dictionary to dictionary. But the assumption of a finite number of discrete senses is quite problematic for natural languages. Often the various senses are actually related to one another and it is unclear where to draw a line between them.

Another difficulty researchers face is the *granularity* of sense distinctions that needs to be taken into account. One might expect the major distinctions between word senses to overlap in most dictionaries which would favor a *coarse-grained sense* in order to make results more comparable. Thesaurus categories only provide a rather coarse-grained distinction because the categories correspond to general conceptual classes, such as *animal or body*, which only provide very broad senses. Also, words in very general categories will not easily be disambiguated because they usually have many closely related senses that will not be captured by the thesaurus categories. Depending on the application, however, this level of sense distinction might not be detailed enough. In that case, the more *fine-grained* distinctions also need to be included in order to be able to distinguish senses on a more detailed level.

3.2 Word Sense Disambiguation Approaches in Literatures.

With regard to the approaches or strategies employed, there are three ways to approach the problem of assigning the correct sense(s) to ambiguous words in context: a *knowledge-based approach*, which uses an explicit lexicon (machine

readable dictionary (MRD), thesaurus) or ontology (e.g. WordNet), *corpus-based disambiguation*, where the relevant information about word senses is gathered from training on a large corpus, or, as a third alternative, a *hybrid approach* combining aspects of both of the mentioned.

3.2.1 Knowledge-Based Approaches

Under this approach, disambiguation is carried out using information from an explicit lexicon or knowledge base. The lexicon may be a machine readable dictionary (MRD), or thesaurus. The Dictionaries or the thesaurus provide definitions (and in some cases example sentences) for each word sense, which contain a number of "clue words". The word sense whose description in the dictionary has a higher overlap of words with the descriptions of the words in the context will be chosen. For example, the following sentences are definitions for two different senses of the word "bank" (river edge or financial institution).

a. land along side of river or lake ("river edge")
b. place where money is kept. ("financial institution")

As it can be seen, these definitions contain clue words, such as *river* or *money*, which are associated with the receptive senses of bank given above. Supposing a given input contains bank and river, one can easily select the first sense for the interpretation of the input *bank*. Given an input, these methods generally compute the number of clue words appearing in the input and definitions, as the score for each candidate sense. Then, the sense with the maximum score is selected. A variation of these methods is to normalize the score by the length of the input.

Lesk (Lesk, 1986) was the first to use dictionary definitions to disambiguate ambiguous words. To automatically decide which sense of a word is intended, he counts overlapping content words in the sense definitions of the ambiguous word and in the definitions of context words occurring nearby. The by now classic example mentioned by Lesk is the word cone which can either mean `pine cone' or `ice cream cone'. Suppose that the word preceding cone in a given sentence is pine. If we compare the dictionary definitions of pine and cone, we find an overlap between the two definitions (marked in bold):

- Pine: kind of **evergreen tree**
- Cone: fruit of a certain **evergreen tree**

So if pine occurs in the same context as cone, we can decide by counting definition overlaps that cone is used in the sense of `pine cone' in that occurrence.

Computing every combination of senses using Lesk's idea and seeking the optimal combination with respect to mutual overlap in entry content words, however, is computationally very expensive because of the huge amount of data that needs to be compared. The introduction of simulated annealing in NLP (Cowie et al., 1992) made the approach practically feasible: rather than computing the definition overlap for all possible combinations of senses, the simulated annealing optimization algorithm identifies an approximate solution. Using the Longman Dictionary of Contemporary English (LDOCE) (Procter, 1978) and simulated annealing, Cowie et al. (Cowie et al., 1992) correctly disambiguated 47% of words to the sense level. When choosing a certain sense, a simple count of the number of tokens in common between all the definitions for a given choice of senses was used. But this method prefers longer definitions because more words can contribute to the overlap. Stevenson and Wilks (Stevenson and Wilks, 2001) alternately compute the overlap by normalizing the contribution of a word to the overlap count by the number of words of the definition that contained the overlapping word. A different extension of Lesk's algorithm is described in (Pedersen and Banerjee, 2002).

Research works have been done using existing lexical knowledge sources such as WordNet (Agirre and Rigau, 1996; Resnik, 1995; Richardson and Smeaton 1995), LDOCE (Guthrie et al., 1991), and Roget's International Thesaurus (Yarowsky, 1992). The information in these resources has been used in several ways, for example Wilks and Stevenson (Wilks and Stevenson, 1997), Harley and Glennon (Harly and Glennon, 1997) and McRoy (McRoy, 1992) all use large lexicons (generally machine readable dictionaries) and the information associated with the senses (such as part-of-speech tags, topical guides and selectional preferences) to indicate the correct sense. Another approach is to treat the text as an unordered *bag of words* where similarity measures are calculated by looking at the semantic similarity

(as measured from the knowledge source) between all the words in the window regardless of their positions, as was used by Yarowsky (Yarowsky, 1992).

The major problem with using MRDs is that dictionaries are created for human use, and due to inconsistencies automatic extraction of large knowledge-bases from MRDs has not fully been achieved so far. Regardless of these shortcomings, MRDs are widely used in WSD for English and provide a ready-made source of information about word senses.

Lately, the use of WordNet as an ontology for WSD has become increasingly popular. WordNet includes various potential sources of information, such as definitions and glosses of word senses, synsets which subsume synonyms representing a single lexical concept and are organized in a conceptual hierarchy, semantic relations (hyponymy and hyperonymy, antonymy, meronymy) between words/synsets. The fact that WordNet provides the broadest set of lexical information in a single resource is one of the reasons for its wide-spread use. Another important characteristic is that it is the first broad-coverage lexical resource that is freely and widely available. WordNet has its limitations as well: its fine-grained sense distinctions and the irregular and varying relative granularity pose a problem often cited in literature. WordNet's sense division and lexical relations have nonetheless become a standard for English WSD.

3.2.2 Corpus-Based Approaches

This approach attempts to disambiguate words using information which is gained by training on some corpus, rather than taking it directly from an explicit knowledge source. This training can be carried out either on a *disambiguated* or on a *raw* corpus, where a disambiguated corpus is one where the semantics of each polysemous lexical item is marked and a raw corpus one without such marking. A corpus-based approach extracts information on word senses from a large annotated data collection, a so called sense-tagged corpus. The possible means used to attribute senses to ambiguous words are then distributional information, context, and further knowledge that has either been annotated in the corpus or added during pre-processing. The corpus-based method approach has the advantage that text material is

easily accessible. The possible means used to attribute senses to ambiguous words are then *distributional information* and *context words*. Distributional information about an ambiguous word refers to using the distribution of senses in a given corpus. Context is composed of the words found to the right and/or the left of a certain word, thus collocational or co-occurrence information. Additional knowledge sources can be exploited, such as lemmas, part-of-speech (PoS), syntactic annotations, etc. Examples of corpus-based systems are (Ng and Lee, 1996), and (Yarowsky, 1993) because performance is usually very accurate and more sense-tagged material is hardly becoming available in the context of common evaluation exercises.

The major difficulty of a corpus-based approach, however, remains the data acquisition bottleneck (Gale et al., 1992b; Ng and Lee, 1996). Raw corpora do not indicate which sense is applicable for a word in a given context. In order to be able to use corpora as an information resource for WSD, they have to be annotated with word senses and this process is very labor intensive. So far, there has not been a lot of sense-tagged material made publicly available, especially for languages other than English. So, one approach to solve the problem has been to manually sense-tag corpora using a given sense inventory, e.g. (Euro)WordNet hierarchies or dictionary sense listings. Another, less time consuming, possibility is the application of less data-intensive (with respect to annotated data) approaches to WSD, such as bootstrapping or unsupervised

Bilingual corpora. Based on the observation that different senses for an ambiguous word in a given language can correspond to distinct words in other languages. Dagan and Itai (Dagan and Itai, 1994) used bilingual corpora for word sense disambiguation. They use Hebrew-English and German-English language pair corpora, respectively). In this case, word polysemy in the source language is defined based on the existence of separate translations in the target language. The objective of this research can be seen as translating a source language sentence to a target language sentence on a word-to-word basis. For example, the Hebrew tuple, "higdil sikkuy" (verb-obj) contains the polysemous word *higdil* is associated with three English tuples: "increase chance", "enlarge chance" and "magnify chance". However, the polysemy can be resolved by selecting the tuple, which is most likely to occur, based on the target corpus, i.e. "increase chance".

Corpus-based WSD systems can be classified depending on how learning is handle: *Supervised Learning*, *Unsupervised Learning* and *semi-supervised Learning* approaches.

3.2.2.1 Supervised Learning Approach

Supervised approaches use annotated training data and basically amount to a classification task. During training on a disambiguation corpus, probabilistic information or statistical information from context words as well as distributional information about the different senses of an ambiguous word are collected. In the testing phase, the sense with the highest probability computed on the basis of the training data is chosen. Training and evaluating such as an algorithm presuppose the existence of sense-tagged corpora. The advantage of this approach is that it yields high accuracy since the decision of selected sense bases on the highest conditional probability. The major difficulties of a corpus-based approach are the need for manual sense-tagging and data sparseness.

So far there has not been a lot of sense-tagged material made publicly available for English, and even for Thai, the corpora are still very small. One approach to solve the problem is to manually sense-tag corpora using for example, WordNet hierarchies (George et al., 1993).

The difficulty of data sparseness for WSD lies in the fact that there is a diverse in frequency among different senses of an ambiguous word. Class-based (Yarowsky, 1992) and similarity-based (Karov and Elderman, 1998).

Depending on the machine learning (ML) algorithm used, corpus-based supervised WSD systems can roughly be classified into *exemplar-based*, *rule-based*, and *probabilistic* approaches.

Exemplar-based paradigm, the k-nearest neighbor technique has been employed most (Dini et al., 2000; Fujii, 1998; Federici et al., 1999; Ng and Lee, 1996; Hoste et al., 2002). The basic intuition behind the systems based on this method is that, because of the distribution of linguistic events with low-frequency events, all

information needs to be taken into account and other learning algorithms are at a disadvantage because they prune training examples that may be useful models to extrapolate from (Daelemans et al., 1999). Therefore, all instances encountered during training are stored in a database and test instances are disambiguated by extrapolating the class of the nearest neighbors contained in the database.

Rule-based approaches (Li et al., 1995; Martinez et al., 2002; Pedersen, 2002; Yarowsky, 2000) use algorithms, e.g. decision lists, which search for discriminatory features in the training data and build an ordered set of rules on the basis of the discriminatory power of these features. The rules are then applied to the test instances. The detail of decision list can be described in below.

Decision Lists. Decision lists are a form of rule representation as proposed by Rivest (Rivest, 1987) and consist of tuples of the form (*condition, value*). As Rivest observed, decision lists can be seen as "if-then-else" rules, in other words, exceptional conditions appear earlier while general conditions appear late in the list. The last condition accepts all cases ("true"), otherwise the system could fail to make any decision for containing input types. Given a query, each condition in the decision list is applied sequentially until a condition which is satisfied by the query is found. Therefore, the value which corresponds to that condition is selected as the answer.

Yarowsky (Yarowsky, 1994) applied decision lists to the task of accent restoration (a single word is associated with multiple pronunciations). (Yarowsky, 1995; Kanokrattanukul 2001) applied the decision lists to the task of word sense disambiguation in English and Thai respectively. In Yarowsky's cases, each condition corresponds to a word collocation, which can be used as evidence to resolve lexical ambiguity and each value corresponds to a correct word sense (or pronunciation). Since manual identification of effective conditions is expensive and inconsistent, Yarowsky used word collocation, within a fixed window size, obtained from a large corpus to automatically identify effective evidence types. The effective degree of a given piece of evidence is estimated as the likelihood that it supports a given sense candidate more strongly than another.

Probabilistic approaches. The third technique is the use of different probabilistic classifiers. Despite its relative simplicity, decision trees and naive Bayes has been frequently applied in WSD with good results (Chodorow et al., 2000; Gale et al., 1992d; Leacock et al., 1998; Mooney 1996; Pedersen, 2000; Pedersen and Bruce, 1997a). Various sorts of log-linear models have also been introduced (Bruce and Wiebe, 1994; Pedersen and Bruce, 1997b; Pedersen et al., 1997; Pedersen, 1998) with success. Lately, combining various probabilistic classifiers has been tested in order to reach better results (Escudero et al., 2000; Florian et al., 2002; Hoste et al., 2002; Klein et al., 2002).

Naïve Bayes Classification. A Bayes classifier uses only distributional information and context words to compute probabilities which corresponds to only using the information which is available from the corpus itself without the need of any additional material, such as a dictionary.

First the disambiguation algorithm is trained on part of the unambiguous corpus, attributing probabilities to the context words found to the right and to the left of the words for various context window sizes. This is done using the Bayes rule

$$P(s_k|c) = \frac{P(c \mid s_k)}{P(c)} P(s_k) \qquad (3.1)$$

Where s_k is the sense k of an ambiguous word w in the context c = $\{c_1,\ldots,c_n\}$, the context words within the specified context window. Since we are only interested in choosing the correct class, the classification task can be simplified by eliminating $P(c)$ which remains a constant for all senses and therefore does not influence what the best class is. Training as used here amounts to count which senses are used most often in a given context.

The context words are assumed to be independent of their position and of each other. They constitute *a bag of words* which corresponds to the Bayes independence assumption. This corresponds to the Bays independence assumption:

$$P(c_1 \ldots c_n|s_k) = P(c_1|s_k)P(c_2|s_k)\ldots P(c_n|s_k) \qquad (3.2)$$

Although the words are not independent of each other, the simplifying assumption allows to adopt an effective model which leads to decisions that can still be optimal even if the probability estimates are inaccurate due to dependencies between features (Domingos and Pazzani, 1997).

Testing takes place on the ambiguous text where the algorithm selects the most probable sense words for each pseudoword according to the Bayes decision rule.

$$\begin{aligned} \text{Decide } s' \text{ if } s' &= \text{argmax}(s_k)\, P(s_k|c) \\ &= \text{argmax}(s_k)\, \frac{P(c \mid s_k)}{P(c)} P(s_k) \\ &= \text{argmax}(s_k)\, P(c|s_k)P(s_k) \\ &= \text{argmax}(s_k)\, P(c_1|s_k)\ldots P(c_n|s_k)P(s_k) \end{aligned} \quad (3.3)$$

Finally, the computed sense words are compared to the original sense words in the disambiguated corpus and the percentage of correctly disambiguated instances is calculated. Despite its relatively "naive" approach, the naive Bayes classifier performs relatively well, especially in comparison with other, more sophisticated approaches (Gale et al. 1993; Mooney, 1996; Escudero et al., 2000).

Decision Trees. In the decision tree algorithm, classification rules are formulated by recursively partitioning the training data. Each nested partition is based on the feature value that provides the greatest increase in the information gain ratio for the current partition. The final partitions correspond to a set of classification rules where the antecedent of each rule is a conjunction of the feature values used to form the corresponding partition. The success of classification using decision trees depends heavily on the attributes or features, which represent the concept of category. Therefore, the feature selection process is the most important part, when we use a decision tree as a classification algorithm. One problem of the decision tree algorithm is the overfitting, where the tree is too specialized to the training data. Researchers have observed that the variance of training data can be reduced by constructing many decision trees using sampling technique.

There are a number of proposal decision tree algorithms "C4.5" (Quinlan, 1993) has been used relatively commonly as a benchmark comparison.

Mooney (Mooney, 1996) and Pedersen and Bruce (Pedersen and Bruce, 1997a) compared the performance of various word sense disambiguation methods with the C4.5 algorithm.

3.2.2.2 Unsupervised Learning Approach

A number of experiments have shown that the performance of word sense disambiguation can be significantly improved by enhancing the volume of supervised learning approach (Mooney, 1996; Hwee, 1997). The supervised method, however, requires considerable manual annotation in supervising large-sized training data sets. To resolve problem, unsupervised methods that do not need annotated training data and allows one to scale to newer and bigger domains easily and quickly have been variously explored.

Since the supervised learning approaches train a model by presenting it with some number of manually created senses tagged examples for a particular. After training, these models are able to assign one of a predefined set of meanings to newly encountered instances of a word. Unsupervised algorithms are applied to raw text material and annotated data is only needed for evaluation. They correspond to a clustering task rather than a classification (or sense tagging) task. Sense tagging is not possible in a completely unsupervised way since it requires that some characterization of the senses be provided. Disambiguation as *word sense discrimination* can be achieved through unsupervised clustering: cluster the contexts of an ambiguous word into a number of groups and discriminate between them without labeling them (Pedersen and Bruce, 1997a; Schütze, 1998). A clear disadvantage is that, so far, the performance of unsupervised systems lies a lot lower than that of supervised systems (Escudero et al., 2000).

The following discussion pays particular attention to Similarity-based methods and automatic clustering method which were earlier discrimination work by Schütze and by Pedersen and Bruce. Pedersen and Bruce (Pedersen and Bruce, 1997b) explored the use of similarity spaces and first order features, while Schütze (Schütze, 1992; Schütze, 1998) developed an approach based on vector spaces and second order.

Similarity-based Methods

The similarity-based method is the computation of *similarity* between an input (a new problem) and examples in the training data (previous problems). In other words, the similarity-based method is defined in term of the method of computing the similarity (or distance).

Suppose each training example is represented by a feature vector. Each example is positioned in an N-dimensional space, where feature f_i corresponds to i-axis. This feature vector is represented by *vector space model* (VSM) that computes the similarity between two examples by angle between the two vector representing the examples. Note that VSM in Information Retrieval (IR) and Text Classification (TC) is used to compute the similarity between documents, which is represented by a vector comprising statistical factors of content words in each document. The similarity between two examples x and y is computed as the cosine of the angle between their associated vectors. This can be expressed by following equation:

$$sim(x,y) = \frac{\vec{x} \cdot \vec{y}}{|\vec{x}||\vec{y}|} \qquad (3.4)$$

where $\vec{x}$ and $\vec{y}$ are vector representing examples x and y respectively

Schütze (Schütze, 1992; Schütze, 1998) applies the vector space model to word sense disambiguation, although vectors are used for each word sense not individual example (In IR/TC, this approach is called *text-to-category* comparison, and contrasted with *text-to-text* comparison which computes the similarity between the input and individual examples. First, Schütze represents each word by a "word vector" that is a vector comprising the statistics (e.g. frequency) of its collecting words. The collocational statistics are usually collected within a fixed proximity. Then each context is represented by a context vector, which is the sum of word vectors related to words appearing in the context. This is unlike the vector space model in Information Retrieval/Text classification. Schütze's method returns a positive similarity value even when two given context vectors have no words in

common. That is two context vectors can be similar when they comprise similar word vectors. Then, automatic clustering algorithm (Douglass et al., 1992) are used to cluster each polysemous word into word senses which are also represented by *sense vectors*. In practice, Schütze used the *singular value decompositon* (SVD) technique (Berry, 1992; Golub and Loan, 1989) which finds the major axes and reduces the dimension of the vector space. The similarity between the input and each word sense cluster is computed by using equation (3.4) to select the word sense with maximal similarity.

Automatic Clustering

Schütze (Schütze, 1992; Schütze, 1998) reduced manual supervision by using automatic clustering algorithms (Douglass et al., 1992). In the Schütze's approach, words, contexts and senses are represented in a high-dimensional real-valued vector space. Two types of representation can be distinguished: *word vectors* and *context vectors*.

Word Vector

A word w can be represented by a vector in which each component corresponds to a word v occurring in the corpus. The vector components represent frequencies of co-occurrence: the component associated with word v is the number of times that v occurs as a neighbor of w in the corpus. A neighbor is a content word occurring in a context window centered on w. These content words are the informants in this approach. Schütze examines two different ways to choose the vector dimension: a local selection which focuses on the 1,000 most frequent words occurring as neighbors of the ambiguous word and ignores the rest of the corpus; a global selection which chooses the 2,000 most frequent words in the entire corpus. In the global manner, word vectors are computed only for the 20,000 most frequent words of the corpus. To compute the most frequent words of the corpus, stop words are excluded. Stop words are conjunctions, prepositions, articles and other words, which appear often in documents yet alone may contain little meaning.

Context Vectors and Senses

The context of an instance *w* is represented by a vector *x* obtained as the weighted sum of the *word vectors* of *w*'s neighbors. Similar *context vectors* can be seen clusters in vector space. Each cluster represents one sense of an ambiguous word and can be characterized by its mean and covariance matrix. The sense of a new instance *w* is then assigned to the most similar cluster.

Schütze reduces the dimensionality of this feature space using Singular Value Decomposition, which is also employed by related techniques such as Latent Semantic Indexing (Deerwester et al., 1990) and Latent Semantic Analysis (Landauer and Dumais, 1997). SVD has the effect of converting a word level feature space into a concept level semantic space that smoothes the fine distinctions between features that represent similar concepts. Clustering algorithms are used to divide the training data into a certain number of clusters. Then, a human expert examines a small number of examples (from 10 to 20) contained in each cluster, which are applied in determining an appropriate word sense for each cluster. Given an input, the cluster (word sense) with the maximum similarity to the input is selected as the interpretation.

Another example is the work of Pedersen (Pedersen and Bruce, 1997b) used altomatic clustering algorithms relying on McQuitty's similarity analysis (McQuitty, 1966) and Ward's minimum-variance method (Ward, 1963). Their training and test set includes polysemous nouns, verbs and adjectives collected from the ACL/DCI Wall Street Journal Corpus (Mitchell et al., 1993) in which each word is annotated with a single sense defined in LDOCE (Procter, 1978) or WordNet (George et al., 1993). What it did have access to was the number of senses for each word and each algorithm split the instances of each word into the appropriate number of clusters. These clusters were then mapped onto the closest sense from the appropriate lexicon. The results were reported that 65-66% correct disambiguation depending on the learning algorithm used. They also tested the expectation maximization (EM) algorithm (Dempster et al., 1977) for unsupervised learning, which resulted in an accuracy of about 63%.

3.2.2.3 Semi-Supervised Learning Approach

Semi-Supervised learning is an algorithm that uses a combination of labeled and unlabelled data. The motivation of semi-supervised learning is that labeled data is expensive to generate while unlabeled data can usually be acquired cheaply. Semi-supervised learning combines labeled and unlabeled data during training to improve performance. The idea behind semi-supervised learning is to exploit the labeled data (supervised learning) to acquire information about the problem and then uses that information to guide learning from the unlabeled data (unsupervised learning). In WSD, example of algorithms that are semi-supervised learning are (Yarowsky, 1995; Park et al., 2002). Yarowsky completely excluded manual supervision by automatically acquiring the initial training data set from a dictionary. Yarowsky used discourse constraints to exclude noise from the decision list. When significantly large number of examples associated with a given discourse are annotated with a common sense in the training data, all examples associated with that discourse are standardized to the same sense annotation. Yarowsky's experimental results show that the performance of this method is equivalent to that achieved by supervised learning. Park (Park et al., 2002) used selective sampling with committees of decision trees. The committee members are trained on a small set of labeled examples which are then augmented by a large number of unlabeled examples. The labels of unlabeled examples can be estimated by using committees. This approach achieved an accuracy improvement up to 20.2% by using unlabeled examples.

Bootstrapping Approaches

The basis of *bootstrapping* is given an initial training data set that usually consists of a small number of annotated examples to progressively enhance the training data by iteratively acquiring presumably correctly annotated examples from previous disambiguation results. To avoid noise data, which comes from incorrectly, annotated examples, a relatively small number of supervised examples is used as initial training data (Hearst, 1991). Karov and Elderman (Karov and Elderman, 1998) used bootstrapping to automatically enhance word sense classifiers using dictionary definitions and a corpus.

3.3 Baseline Evaluation

3.3.1 An Upper bound Performance

An upper bound performance is the disambiguation performed by a human. In case of word sense disambiguation, if a human cannot disambiguate correctly, it is expected that a machine cannot either. The case that human cannot perform correctly is that the information in the context is not enough. Gale (Gale et.al, 1992a) found that in disambiguating words which have no related meanings (homonyms such as bank) has upper bound 95% or higher. In disambiguating words that have highly related meanings (polysemous such as title) has upper bound only 65-70%.

3.3.2 A Lower bound Performance

A lower bound performance is the performance of the simplest algorithm. For example, assuming that an ambiguous word occurs 1000 times in a corpus, with 600 times of sense1, 200 times of sense2, and 200 times of sense3. If choosing the most dominant meaning in all cases, the algorithm will achieve the accuracy rate of 60%

3.3.3 Baseline

As a baseline system, the most frequent sense (MFS) of a word is chosen as the correct sense. This means that the baseline system always return the sense that has the highest frequency in a corpus. The frequency of word senses is calculated from the occurrences of the word senses in the corpus and assign to most frequent sense. The baseline system can be computed as follows.

$$\text{Baseline (MFS)} = \frac{\text{Frequency of } S_i \text{ in the corpus} * 100}{\text{Total number of answered senses}} \qquad (3.5)$$

Where S_i is the sense that has the highest frequency in the corpus. Total number of answered senses is the same as the number of training data.

3.4 Our Work Approach

In this chapter, related works in the word sense disambiguation problem are presented. Our methodology which is presented in this thesis is based on distributional semantics method as the unsupervised learning approach to solve the problem of word sense disambiguation. The unsupervised learning approach corresponds to a clustering task rather than a classification (or sense tagging) task.

In the distributional semantics method, words, contexts and senses are represented in real-valued vector space. The words that are used in similar contexts will have the same or a closely related meaning. Our approach is the task of grouping the meaning of the target word that is used in similar contexts. The detail of distributional semantics method will be described in the next chapter.