An Insight into Word Sense Disambiguation Techniques

Harsimran Singh
University Institute of Engineering and Technology,
Panjab University, Chandigarh, India

Vishal Gupta
Assistant Professor,
University Institute of Engineering and Technology,
Panjab University, Chandigarh, India

ABSTRACT
This paper presents various techniques used in the area of Word Sense Disambiguation (WSD). There are a number of techniques such as: Knowledge based approaches, which use the knowledge encoded in Lexical resources; Supervised Machine Leaning methods in which the classifier is made to learn from previously semantically annotated corpus; Unsupervised approaches that form cluster occurrences of words. Then there are also semi supervised approaches which use semi annotated corpus as reference data along with unlabeled data.

Index terms: Word Sense Disambiguation, Natural Language Processing, WordNet, supervised, unsupervised, semi-supervised.

1. INTRODUCTION
Word Sense Disambiguation is the determination of sense of a particular word used in a particular context. Since words can have multiple senses in dictionary, it is very much important for the machine to determine the correct meaning of word in the context. For instance consider the following:

a) He gave me his own **pen** for writing paper.
b) You must **pen** down all the instructions of the teacher.

Now in the above sentences two different meanings of pen are denoted: an instrument of writing and to write something respectively.

It is a collusive problem of Computer Linguistics and Artificial Intelligence. It is unique feature that a particular sense of word is triggered by the use of the word in a sentence or in a particular context. Human brain does is automatically through perceived intelligence and knowledge repository which he has developed through learning and most of time human brain does not of these lexical ambiguities of language. But making a computer to disambiguate is main crux of problem. This computational recognition of senses of word is what is known as Word Sense Disambiguation (WSD). Hence such a system would sense tag above two sentences as:

a) He gave me his own **pen**/***writing instrument*** for writing paper.
b) You must **pen**/***write*** down all the instructions of the teacher.

WSD can either be considered as a stand- alone problem or as an integrated part of other language processing tasks such as text summarization, machine translation, etc. One of the forms of knowledge sources can be corpora of text which is either semantically annotated (sense tagged) [1] data or raw un-annotated (unlabeled) data. Supervised techniques rely on former while unsupervised techniques do not require sense tagged data and hence they rely on latter. In order to obtain good results, supervised algorithms have to be trained heavily using sense tagged data (which is known as "knowledge acquisition bottleneck") but such a condition is not possible since hand tagged data is expensive to create and after learning the classifiers are only applicable to text that are similar to subject. Various supervised approaches are there such as Bayesian Learning, Decision List [8], Exemplar-Based learning [11] Support Vector Machines [18]. Lack of portability of supervised methods to different languages and due to limitation of using fixed sense repository which constrains us to use only the senses that are present in the repository. Because of this limitation, Unsupervised approach has gained popularity. It is because of its resource consciousness and robustness. These methods work on raw data, making clusters of context where each cluster corresponds to a sense. Much research has been carried out in unsupervised approaches since late 90s. These include HyperLex [5], Yarowsky's approach [3], Parallel word corpora, etc.

Knowledge based approaches are among the oldest disambiguating techniques which came into use in 1970s and 80s. We will discuss them in the following section.

Semi-supervised or minimally supervised approaches lie midway between supervised and unsupervised approaches. The notion behind these approaches is to use unsupervised techniques to form clusters of text which represents the sense of word but since the clusters by themselves do not have a predefined meaning, the idea is first to initially train the classifier with some seed data (using some supervised algorithm) and then classifier is used on untagged corpus to obtain larger training set. These methods have gained popularity because they require a small amount of tagged data while often out performing unsupervised methods for large data. Semi-supervised learning techniques include Bootstrapping sense tagged seed examples [8] and Monosemus Relative. Semi-supervised approaches form a buffer between supervised and unsupervised approaches.

A brief review of all above techniques will be presented in this paper.

[1] Sense tagged means that each occurrence of word, in the text corpus, is manually tagged with most appropriate sense which human thinks is most appropriate

2. KNOWLEDGE BASED DISAMBIGUATION

Knowledge or Dictionary based methods disambiguate senses of the word by finding the semantic similarity between the descriptions of a pair of words in their dictionary definition. The fundamental principle behind this methodology is the matching of information, obtained from the context of the word, with the information, obtained from the lexical knowledge base[2]. These methods rely only on the information provided by dictionaries. Before going in further, let us discuss something about these dictionaries or knowledge bases and see what they actually provide for which they are used for disambiguation. A MRD provides the following things:

- A list of all the meanings of the given word.
- A description or definition of all word meaning.
- One or more example sentences for most of the meanings.

A thesaurus, e.g. *Roget's International Thesaurus* [22] gives the relationships between words such as synonymy, antonymy, etc. In the above example a thesaurus add, for example:

1. flower, bloom, bright
2. flower, plant, flora

A semantic network like WordNet [15] provides other relations of hypernymy/hyponymy (is-a relationship), meronymy/holonymy (part-of relationship), entailment, etc.

2.1 Overlap Based approaches

A simplest knowledge based approach is to find the overlap between the senses definition of the target words in the dictionary. It based on the principle that word occurring in the context tend to share same topic. Hence the terms or words contained in the neighborhood are compared with the dictionary senses of the ambiguous. This method is known as *Lesk algorithm* [2].

Algorithm:

1. Retrieve from MRD, all sense definitions of the words to be disambiguated.
2. Determine the definition overlap for all possible sense combinations
3. Choose senses that lead to highest overlap

For Example[3]: disambiguate PINE CONE

- Pine
  1. kinds of evergreen tree with needle-shaped leaves
  2. waste away through sorrow or illness
- Cone
  1. solid body which narrows to a point
  2. something of this shape whether solid or hollow
  3. fruit of certain evergreen trees.

Pine#1 ∩ Cone#1 = 0
Pine#2 ∩ Cone#1 = 0
Pine#1 ∩ Cone#2 = 1
Pine#2 ∩ Cone#2 = 0
Pine#1 ∩ Cone#3 = 2
Pine#2 ∩ Cone#3 = 0

This method is also known as *gloss overlap.* If $S_1 \in$ Senses(w_1) and $S_2 \in$ Senses(w_2), for a pair of words w_1 and w_2, then score is calculated by below formula:

$$\text{scoreLesk}\ (S_1, S_2) = |\text{gloss}(S_1) \cap \text{gloss}(S_2)|$$

where gloss(S_i) is the bag of words in the textual definition of S_i of w_i.

But since it leads to a large number of combinations and finding optimal sense combination is difficult. This problem is solved by using a Simplified Lesk algorithm proposed by [10]. The idea is take the target word and retrieve all its senses from MRD. Then for each sense definition, find its overlap with the current context. Hence the above formula for calculating the score will change to as:

$$\text{scoreLesk\_sim}\ (S) = |\text{context}(w) \cap \text{gloss}(S)|$$

where context(w) is the bag of words in the context window around target word.

For Example: disambiguate PINE: *"Pine cones hanging in a tree"*. Pine#1 ⋂ Sentence = 1 and Pine#2 ⋂ Sentence = 0. But despite this the Lesk method have inherit limitation of being too dependent to exact wording of the definitions. In order to overcome this limitation extended Lesk algorithm proposed by Banerjee and Pedersen [19]. The meanings of words (i.e., glosses), connected to the words to be disambiguated, through various relationships, are defined in WordNet (hypernymy, hyponymy, meronymy, etc.). This is more effective source of information and improves disambiguation accuracy. Now the score formula will become:

$$score_{ExtLesk} = \sum_{s' \subseteq rel(s) or\ s=s'} |\text{context}(w) \cap \text{gloss}(s')|$$

where rel(s) gives the sense related to s in WordNet under some relations.

Consider an example: "On combustion of coal we get ash". In this sentence **ash** cannot be disambiguated from previous technique since there is no overlap. By applying the extended Lesk approach the residue sense of **ash** becomes winner because of the overlap between the fly ash descriptions, which comes from hyponymy relation, and the sentence due to word combustion. This deeper level overlap helps in distinguishing the meaning. But this method also has a critique that due to increased region of matching in WordNet there are increased chances of topic drift.

The accuracy of Lesk algorithm was initially found out to 18.3% Extended Lesk: 34.6%.

Walker's algorithm [1] is another overlap based approach. It relies on Thesaurus. In this approach we first find, for each

[2] Knowledge base can be Thesaurus, Machine Readable Dictionaries (MRD) like the Longman Dictionary of Contemporary English (LDOCE) [24], the Oxford Dictionary of English [20], Ontologies, etc.

[3] Example is from [2]

sense of target word, the thesaurus category to which that sense belongs. Then a score is calculated for each sense using the words in the context. The score for a sense is incremented by 1, for each context word, if the thesaurus category of the word matches with that of the sense. For example: "The **chair** was very comfortable for sitting though it was costly". Now target word is "chair" and it has two senses: furniture sense and presider of meeting sense i.e. chairperson. Clue words from the context are: comfortable, sitting, and costly. From table 1 it is clear that furniture sense is winner sense and it is appropriate here.

Table 1: Above example.

	Sense1: furniture	Sense2: presider
comfortable	+1	0
Sitting	+1	+1
Costly	+1	0
Total	**+3**	**+1**

2.2 Selectional Preferences

The main principle behind *Selection preferences or restrictions* is to restrict the meanings of the word in a given context. Basically it is the restriction or constraint which a word imposes on its argument or context (usually through grammatical relationships). For example: the meaning of word **eats** appears to imply that its direct object must be edible entities. Selectional preference dates back to an ancient Indian tradition of WSD. Some words in the sentence have a "Desire" ("*aakaangksha*"), like in the sentence "I saw a boy with a long hair". Here the verb "*saw*" and the noun "*boy*" desires an object. Some words have "Appropriateness" ("*yogyataa*") to fill this desire. Proximity" ("*sannidhi*") can determine the meaning. For example: "I saw a boy with telescope". In this case prepositional phrase "*with the telescope*" can be attached to both "*boy*" and "*saw*" but it is attached to "*boy*" due to proximity check.

Finding selection preferences is finding a relationship between entities. These entities can be words or the semantic classes. Hence there can be word-to-word relations, word to class relations, and/or class to class relations. Selectional preference is basically finding the appropriateness ("*yogyataa*") of word-to-word relation. Simplest measure for selectional preference is *frequency count*. Mathematically it is expressed as: Count (w_1, w_2, R) [Here R is relation connecting to words w_1 and w_2]. Another measure of semantic fit between words is *conditional probability*:

$$P(W_1|W_2,R) = \frac{Count(W_1,W_2,R)}{Count(W_2,R)}$$

Resnik [16] proposed a selectional preference method based on combining statistical and knowledge based approaches. He proposed the difference between prior distribution and posterior distribution determines the selectional preference. For example: prior probability of <person> may be higher than that of <insect> but when the identity of predicate is taken into consideration (if the verb is "*buzz"*) then the probability of <insect> becomes more. Given a word *W* and semantic class *c* such that relation R exists between them, then selectional preference strength of a predicate is given by:

$$S_R(W) = \sum_c P(c|W,R)\log\frac{P(c|W,R)}{P(c)}$$

Selection Association is given by:

$$A_R(W,c) = \frac{1}{S_R(W)}P(c|W,R)\log\frac{P(c|W,R)}{P(c)}$$

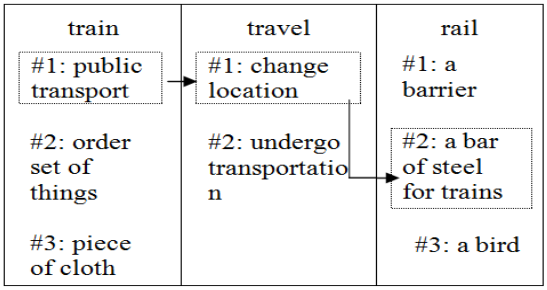
where $P(c|W,R) = \frac{Count(W,\ c,\ R)}{Count(W,\ R)}$ and $Count(W,\ c,\ R) = \sum_{w'=c}\frac{Count(W,\ W',\ R)}{Count(W')}$

Resnik assumes an equal sense distribution. This is word-to-class relation model. Class-to-class model was given by Agirre and Martinez [13]. Average precision and coverage, for 8 nouns[4] was found out to be 95.6% and 26% for class-to-class. In case of the other two it was 66.9% and 86.7%, and 66.6% and 97.3% respectively.

2.3 Semantic similarity

Often it is human behavior that we use those words in the discourse which effectively communicates our dialogue. This unique property of human language is an important constraint which could be exploited for WSD. Discourse is coherent [23] if the words in the discourse are related. Semantic similarity method follows this principle. Appropriate senses of the words (which are in context) can be determined by finding the semantic distance between different senses of words (in context) and choosing those which are at least distance. It is important to point out that the context, talked above, can be **local** or **global**. In local context, the window is restricted to few words around the target word. And in global context it consists of lexical chains which are sequence of semantically related words and corresponds to coherence of discourse. For example: "A very long **train** travelling along the **rails** at a constant **velocity** v in a certain **direction**"

Table 2: Lexical chain for "train", "travel" and "rail"



The meaning of the words is identified based on their membership of lexical chain.

Most of the similarity based methods takes input as concepts and yield a value which gives the semantic affinity among concepts. [6] Give a formula in which semantic score is calculated by finding the minimum path length between two concepts in the semantic network given by:

$$Similarity(C_1,C_2) = -\log\frac{Path(C_1,C_2)}{2D}$$

where $Path(C_1,C_2)$ is the distance between concepts and D is the depth of taxonomic structure.

Based on this Resnik [17]suggests a measure of similarity by calculating the Least Common Subsumer (LCS) which is the

[4] Current performance parameters are for verb-object pair.

first common node encountered in hierarchal network while going from one concept to other, for a pair of concepts.

$$Similarity(C_1, C_2) = IC\big(LCS(C_1, C_2)\big)$$

where IC is information content.

Agirre and Rigau [9] gave distance based measure for calculating the similarity by introducing the measure of Conceptual Density (CD). Conceptual density is the measure for determining the closeness in meaning among pairs of words, taking reference a structural hierarchical net, say WordNet. For a given concept c, the conceptual density is given by:

$$CD(c, m) = \frac{\sum_{i=0}^{m} nhyp^{i^{0.2}}}{descendents_c}$$

where $nhyp$ denotes the mean number of hyponyms per node, m is the number of (marks) senses of words to disambiguate, $descendants_c$ is the height of sub hierarchy and 0.2 is the smoothing factor found experimentally. The density will be highest density for the sub-hierarchy which contains more senses of those, relative to the total amount of senses in the sub-hierarchy.

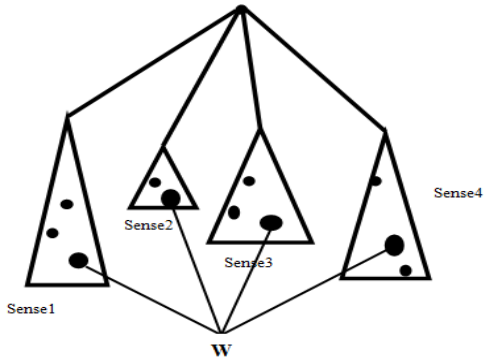


Figure 1: [21] It shows word W, which is to be disambiguated has four senses in the hierarchical network as indicated by four triangles. The other dots in the sub-hierarchies indicate the words in the context of the target word W.

Amoros and Heradio [21] take Agirre and Rigau as baseline and suggests a varied formula which finds out to be 24% more efficient then Agirre and Rigau.

3. SUPERVISED MACHINE LEARNING APPROACHES

In supervised machine learning approaches, a classifier is trained on training data (which is manually sense marked corpus), using machine learning techniques, and then the classifier is used on test data, to see how much accurately it has classified the data. The basic idea is that words around the target word provide clues about the sense of the word, these words are called *features*. These features are learned along with the weightage, in the environment of the word to be disambiguated. And from there when a new sentence comes up we would like to produce disambiguation of the target word. We now discuss some of common supervised machine learning approaches.

Naïve Bayes approach is simple yet very effective machine learning method. In simple words it can be stated as: From among all senses of an ambiguous word, the classifier assigns that sense which maximizes the probability of that sense given the feature set of the target word. Mathematically,

$$s = \boldsymbol{argmax}_{s \in senses} P(s|V_W) \quad \textbf{... eq1}$$

$$= \boldsymbol{argmax}_{s \in senses} P(V_W|s)P(s) \quad \textbf{... eq2}$$

where V_W is the feature vector. It contains part-of-speech of target word, Collocation vector (bag of words around it), Co-occurrence vector (number of times a word occur in the bag of words around it). Eq 2 is obtained by applying Bayes rule and considering naïve independence assumption which states that each feature is independent of all other features in the feature vector.

In spite of independent assumption, it has proven to be good among supervised learning approaches. It gave an average precision of 64.13% in Sensval-3 (All words task).

Decision Lists [8] approach is based on property of 'One sense per collocation' [7]. It means that it means that bag of words surrounding the target word points to one and unique sense of that word.

Algorithm:

1. Collect a large collection of collocations for the target word (training data).
2. For each collocation calculate the word- sense probability distribution.
3. Calculate the log-likelihood ratio as
$$\text{abs}\{\log \frac{P(Sense_i|Collocation_j)}{P(Sense_j|Collocation_j)}\}$$
4. Rank the collocations in the form of a decision list with higher log-likelihood collocations ranked higher.
5. Classify the test sentence based on the highest ranking collocation found in the test sentence.

For example:

Table 3: Example Decision List based some training data

LogL	Collocation	Sense
11.3	river bank	⇒ B
9.78	money (within ±k words)	⇒ A
9.62	water (within ±k words)	⇒ B
9.53	bank account	⇒ A
9.50	financial (within ±k words)	⇒ A
…	…	…

In this example we only consider two senses of word *bank*, financial sense, that is A and side of river sense, that is B. From training examples we obtain log likelihood for each collocation and then we arrange the log likelihoods in the decreasing order to obtain the decision list. Now classification of a test sentence is based on higher ranking collocation found in the test sentence. For example "your initial check should be drawn on your nominated **bank** account …" Since **account** is in the neighborhood of the word **bank** therefore from table 3 it implies that here bank is being used in sense A.

Decision Lists gave precision of 96% when tested on 12 highly polysemous English words.

Exemplar Based Learning [11] also known as instance or memory-based learning, is a simple technique in which the training examples are cumulated in memory, arranged in a feature space. When a new example comes, it is classified based upon its closeness to already stored examples in memory. The most common approach for Exemplar based learning is *k-NN (k- Nearest Neighbor)* in which new example is classified based on similarity measure (e.g., distance functions). It uses similarity measure to find the k nearest cases to new case. An example is classified by a majority vote of its neighbors with example being assigned to the class (here sense) most common amongst its k nearest neighbors measured by a distance function. The distance measure is Hamming Distance:

$$D_H = \sum_{t=1}^{m} |x_{it} - y_t|$$

$$D = 0 \; if \; x_{it} = y_t \; and \; D = 1 \; if \; x_{it} \neq y_t$$

where each stored example sentence $\mathbf{x_i}$ consists of **m** feature values and is represented as $\mathbf{x_i} = (x_{i1}, x_{i2}, x_{i3}, \ldots\ldots x_{im})$ and new example is **y** is represented as $\mathbf{y} = (y_1, y_2, y_3, \ldots\ldots y_m)$. For each new example k nearest classes are selected and it is assigned to the class that is most common among them.

Figure 2: 3-Dimentional Feature Space

Figure 2 shows a three dimensional feature space with two classes. The new instance marked in the figure needs to be assigned a class based on its nearness measure.

Support Vector Machines [18] are linear classifiers which are based notion of designing a hyperplane that classifies all training examples (vectors) in two classes. Since the can be more than one hyperplanes which can classify all the instances so best choice is the hyperplane that leaves the maximum margin from both classes. The hyperplane is defined by an equation as:

$$g(x) = W^T x + b$$

$$g(x) \geq 1, \forall x \in class\ A \; and \; g(x) \leq -1, \forall x \in class\ B$$

where W is the normal vector which determine the orientation of hyperplane and b is bias which controls the displacement of the hyperplane from origin. The width of margin comes out to be $\frac{2}{\|W\|}$ it means that the width is inversely proportional to length of normal vector.

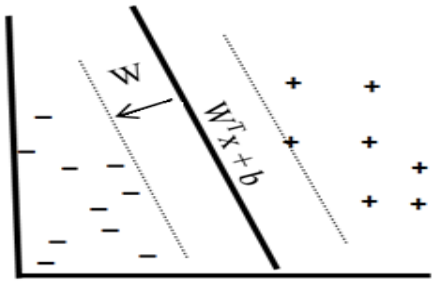


Figure 3: Intuition of SVM

It is important to note that this system works only when the training samples are linearly separable. But if the space contains samples which are linearly inseparable then we transform the space into more convenient space. This is done using the kernel function K. This function (K) allows avoids knowledge of transformation (∅) into another space: $K(x_i, x_j) = \emptyset(x_i) \cdot \emptyset(x_j)$

SVM gave precision and recall of 72.4%in Sensval-3 (Lexical sample task, used for disambiguation of 57 words).

4. SEMI-SUPERVISED MACHINE LEARNING APPROACHES

Basic idea of semi-supervised approaches is the following:

1. Train the machine using seed data.
2. Tag the unseen data.
3. Manually correct the tags.
4. Retrain using larger data.
5. Repeat steps 3 and 4 until satisfactory accuracy level is reached.

These are also known as minimally supervised approaches. Fully labeled data is often not available and it is expensive, laborious, and time consuming to label the unlabeled data. Therefore in these cases semi-supervised becomes very viable. Semi-supervised approaches try to solve this problem by using large amount of unlabeled data, which is easy to collect but there are few ways to use them, along with labeled data, to build better classifiers. We now present two such approaches.

Bootstrapping. It starts with small amount of sense labeled data, a large amount of unlabeled data and one or more classifiers. Yarowsky's bootstrapping algorithm [8] which uses Yarowsky's supervised algorithm that uses Decision Lists. It makes two assumptions:

1. One sense per Collocation: [7] the words around the target word provide a strong clue about the sense of that word.
2. One sense per Discourse: [23] within a given discourse or document, only one particular sense of the word is being referred.

The algorithm as follows:

1. Identify all the examples of the given polysemous word and for each possible sense of the word, identify relatively small number of training examples representative of that sense.

2. Train the *Decision List* algorithm using a small amount of seed data.
3. Classify the entire sample set using the trained classifier.
4. Create new seed data by adding those members which are tagged as Sense-A or Sense-B with high probability.
5. Retrain the classifier using the increased seed data.

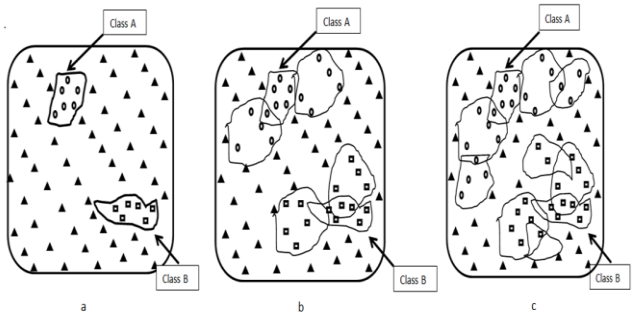


Figure 4: Yarowsky's Bootstrapping algorithm example. (a) Initial seed data. (b)(c) Growth of seed set.

This approach gave same precision as Supervised Decision List i.e. 96% but the extra amount of work required on annotating the corpus is not required.

Monosemus Relatives. The idea here is to use web as a corpus to build annotated data sets. Since web has grown in various dimensions at an unbelievable rate, various approaches are being explored to use huge sized web as an annotated corpus.

For each word *w*, the words which are synonymous to *w* and have single sense are found. Now for each such word is web search and its contexts are found. Then these contexts are directly sense tagged with the sense of the word to create sense annotated corpus.

5. UNSUPERVISED MACHINE LEARNING APPROACHES

In supervised learning, each example is labeled with a particular sense. In other words whether the financial sense or the river sense of the word *bank*, is being used in a sentence. So for each example in supervised learning we explicitly want, so called right answer. But in unsupervised learning, dataset have is no label. Given this unlabeled data, the unsupervised learning algorithm finds the structure in the data.

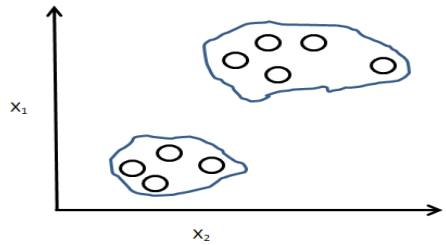


Figure 5: Unsupervised Learning algorithm action on a data set (it forms cluster of data).

Unsupervised algorithms are known as Clustering algorithms and are used in many places e.g. Google News.

They can overcome knowledge acquisition bottleneck since they do not require sense annotated data. But on the other hand they form cluster which might not be equivalent to sense in dictionary. Thus they require manual checking to determine how members of each cluster are related to each other. We now discuss some of the common unsupervised machine learning approaches.

HyperLex [5] is a graph based unsupervised WSD technique. The algorithm works in three steps. In first step, a cooccurrence graph G = (V, E) is constructed using the words as vertices that occur in paragraphs of the text in which the target word occurs. There is an edge between two words if those words occur in the same paragraph[5]. Each edge has a weight which is given by:

$$W_{A,B} = 1 - \max\{ P(A|B), P(B|A) \}$$

where $P(A \mid B) = f(A,B) \,/\, f(B)$, and $P(B \mid A) = f(A,B) \,/\, f(A)$ and f stands for frequency of words[6].

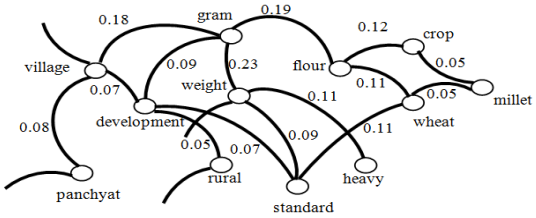


Figure 6: Cooccurrence graph for the word "gram"

In the second step a Minimum Spanning Tree (MST) is generated from this graph. First root hubs are identified which are nodes with high relative degree. All these root hubs are connected to the target word using an edge of zero weight to form MST. This set of root hubs is the sense set of the target word.

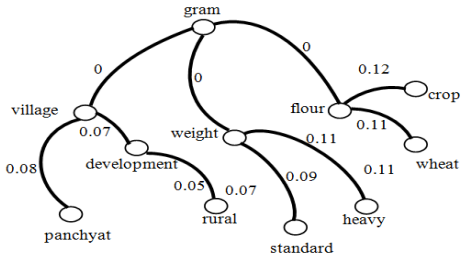


Figure 7: MST generated for the word "gram"

In the final step, MST is used to find the specific sense of the target word. Each node in MST is assigned a score s which is given by:

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

where $d(h_{i,j})$ is the root hub h_i and node j in the MST.

[5] Only nouns and adjectives are considered in experimentation while verbs and adverbs were ignored
[6] Words with occurrence ≥ 10 are considered.

For a given context, score all the words occuring in the context are cumulated. The component with the maximum score is the winner sense.

HyperLex gave precision of 96% when tested on 10 highly polysemous French words.

WSD using Roget's thesaurus Categories. Yarowsky [3] presented an approach which exploited the features provided by Roget's Thesaurus [22] for disambiguating the sense of the word. Roget's Thesaurus provides categories or class in which a word can fall. For example [3]: the word ***crane*** may fall in the category of MACHINE/TOOLS or in the category of ANIMAL/INSECT. The algorithm is based on observations that when a word appears in a particular context, it has a particular sense and that particular sense belongs to a particular thesaurus category. So a discriminator which can discriminate conceptual classes of thesaurus can also be used to discriminate the members of those classes. The algorithm as follows:

1. Collect context which are representative of Roget category.
2. Identify and weigh the salient words for each category as:

$$Weight_w = P(w|RogCat)/P(w)$$

where $P(w|RogCat)$ is the probability of the word w appearing in the context of Roget category $RogCat$ and $P(w)$ is the overall probability of the word in the corpus.

3. Predict the appropriate category for the target word from the above computed weights as:

$$\underset{RogCat}{argmax} \sum_{w \in context} \log \frac{P(w|RogCat) \times P(RogCat)}{P(w)}$$

This approach gave a precision of 92% when tested on 12 highly polysemous words.

Lin's approach for WSD. Lin [14] gave a clustering based WSD technique. The approach was based on the intuition that different words, occurring in the similar local context are likely to have similar meaning. For example consider the sentence from [14]: "The facility will employ 500 of the existing 600 employees."

The word *facility* has 5 possible senses: installation, proficiency, adeptness, readiness, and toilet. In order to disambiguate the word, we consider the words, from the text corpus, which occurred in the identical context of the word *facility*. Here the context word is *employ* so we consider all words which have *employ* word in their context. Table 3 shows such words along with their frequency and log likelihood. From the table it is clear that here the sense for *facility* is *installation.*

The algorithm for Lin's approach is as follows:

1. The input sentence, containing the ambiguous word w is parsed to extract the local context for the target word.
2. Search is made through local context database to find the words S_w (known as *Selectors* of w) which has identical context as the word w.
3. Choose the sense that maximizes the similarity between S_w and w and assign the selected sense to all occurrences of the target word.

The similarity function is given by:

$$sim(A,B) = \frac{\log P\big(common(A,B)\big)}{\log P\big(describe(A,B)\big)}$$

where $\log P\big(common(A,B)\big)$ is the amount of information in the commonality of A and B, and $\log P\big(describe(A,B)\big)$ is the information needed to fully describe A and B.

Table 4: Subjects of the word "employ" from [14]

Word	Frequency	Log Likelihood
ORG	64	50.4
Plant	14	31.0
Company	27	28.6
Industry	9	14.6
Unit	9	9.32
…	…	…

Lin's approach gave a precision of 68.5% when tested on 7 files of SemCor containing 2832 polysemous nouns.

6. REFERENCES

[1] Walker D. and Amsler R. *The Use of Machine Readable Dictionaries in Sublanguage Analysis* in Analyzing Language in Restricted Domains, Grishman and Kittredge (eds), LEA Press, pp. 69-83, 1986

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

[3] Yarowsky D. *Word sense disambiguation using statistical models of Roget's categories trained on large corpora* in Proceedings of the 14th International Conference on Computational Linguistics (COLING), Nantes, France, 454-460, 1992

[4] Lin D. *Using syntactic dependency as local context to resolve word sense ambiguity* in Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics (ACL), Madrid, 64-71,1997.

[5] Vacronis J. *HyperLex: Lexical cartography for information retrieval* Computer Speech & Language, 18(3):223-252, 2004.

[6] Leacock, C. and Chodrow, M. 1998. *Combining local context and WordNet similarity for word sense identification*. In WordNet: An electronic Lexical Database, C. Fellbaum, Ed. MIT Press, Cambridge, MA, 265–283.

[7] Yarowsky, D. 1993. *One sense per collocation*. In Proceedings of the ARPA Workshop on Human Language Technology (Princeton, NJ). 266–271.

[8] Yarowsky, D. 1994. *Decision lists for lexical ambiguity resolution: Application to accent restoration in Spanish and French*, in Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics (ACL), Las Cruces, U.S.A., 88-95, 1994.

[9] Agirre, E. & German R. 1996. *Word sense disambiguation using conceptual density*, in Proceedings of the 16th International Conference on Computational Linguistics (COLING), Copenhagen, Denmark, 1996.

[10] Vasilescu, F., Langlais P., and Lapalme G. 2004. *Evaluating variants of the Lesk approach for disambiguating words*. In Proceedings of the Conference of Language Resources and Evaluations (LREC 2004).

[11] Aha D. W., Kibler D., and Albert. 1991 M. K. *Instance–based learning algorithms*. Machine Learning, 6(1):37–66.

[12] R. F. Bruce and J. M. Wiebe. 1999. *Decomposable Modeling in Natural Language Processing*. Computational Linguistics, 25(2):195–207.

[13] Agirre, E. and Martinez, D. 2001. *Learning class-to-class selectional preferences*. In Proceedings of the 5th Conference on Computational Natural Language Learning (CoNLL, Toulouse, France). 15–22.

[14] Lin D. *Using syntactic dependency as local context to resolve word sense ambiguity* in Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics (ACL), Madrid, 64-71,1997.

[15] Miller, G. Wordnet: A lexical database. ACM, 38(11) 1995

[16] Resnik, P. *Selection and Information: A Class-Based Approach to Lexical Relationships*. University of Pennsylvania 1993.

[17] Resnik, P. *Using information content to evaluate semantic similarity*. IJCAI 1995.

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

[19] Banerjee, S., and Pedersen, T. 2003. *Extended gloss overlaps as a measure of semantic relatedness*. In Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence, 805–810.

[20] Soanes, C. and Stevenson, A., Eds. 2003. *Oxford Dictionary of English*. Oxford University Press, Oxford, U.K.

[21] Fernandez-Amoros, D., and Heradio, R. *Understanding the role of conceptual relations in Word Sense Disambiguation*, Expert Systems with Applications (38:8) 2011, pp. 9506-9516.

[22] Roget, P. M. 1911. *Roget's International Thesaurus*, 1st ed. Cromwell, New York, NY.

[23] Halliday, M. A. and Hasan, R., Eds. 1976. *Cohesion in English*. Longman Group Ltd, London, U.K.

[24] Proctor, P., Ed. 1978. *Longman Dictionary of Contemporary English*. Longman Group, Harlow, U.K.