Chapter 4

Word Sense Disambiguation

This chapter discusses WSDin great detail. The key approaches to WSD and the related algorithms entailing those approaches are studied in depth to gain insights as to how problems related to WSD can be tackled.

4.1 Introduction

Word Sense Disambiguation (WSD) is one of the cardinal and absolutely critical problems of NLP. It is one of the most amazing feats of human mind that we understand written and spoken communication in spite of enormous number of possibilities that exist because of multiple meanings of words that compose a sentence. It is equally amazing that we produce a correct sentence choosing words in their appropriate context. So any system that proposes to implement NLP on a computer has to address very seriously the question of WSD. WSD can be said to be the Holy Grail of NLP in the sense that if it is solved other hard problems of NLP such as Machine Translation (MT), Information Retrieval (IR) , Question-Answering etc can be successfully tackled to a considerable degree. WSD became more well founded and well understood when the immensely important lexical resource called WordNet was created.

Formally, Word Sense Disambiguation (WSD) is a mechanism to obtain the sense of target words or all words (All word WSD more difficult) against a sense repository like the WordNet or a Thesaurus using the context in which the word appears. A basic axiom of WSD is **"words around the target word (word meant to be**

disambiguated) , known as context words, provide strong and consistent clues for disambiguation".

For e.g. "He went to the **bank** to withdraw some money". Here bank is a polysemous word and the target word having three senses:-

1. Bank (Noun) – financial sense
2. Bank (Noun) – river bank sense
3. Bank (Verb) – depend/rely

The word "bank" in the above sentence needs to be disambiguated by the machine using the context (surrounding words) in which it appears.

4.2 Approaches to WSD

Broadly there are three approaches to WSD. These are as follows:-

1. Knowledge Based Approach:- It relies on Knowledge resources like WordNet , Thesaurus etc. It may use grammar rules as well as hand coded rules for disambiguation.

2. Machine Learning based approach:- It relies on corpus evidence. Here a model is trained using tagged or untagged corpus. Probabilistic and Statistical models are built from the training corpus.

3. Hybrid Approach:- This sort of approach acquires knowledge from diverse sources like WordNet and corpus. It is similar to Supervised approach in the sense that some sense tagging is required here. It is similar to Unsupervised approach in the sense that information in the form of corpus evidence is utilised here. On the whole it can combine information from multiple knowledge sources and use a comparatively small amount of tagged data.

4.2.1 Knowledge Based Approach

Knowledge Based Approaches can be broadly divided into the following categories:-

- WSD Using Selectional Preferences and Arguments
- Overlap Based Approach

4.2.1.1 WSD Using Selectional Preferences and Arguments

WSD Using Selectional Preferences and Arguments comes under Knowledge Based Approach. It requires exhaustive enumeration of argument structure of verbs, selectional preferences of arguments and description of properties of words such that meeting the selectional criteria can be decided.

Sense 1	Sense 2
This airlines **serves** dinner in the evening flight	This airlines **serves** the sector between Agra and Delhi

For the first sense , serves which is a verb has an agent which is airlines and also an object which is edible , whereas for the second sense serves has an agent which is airlines and also an object which is sector. Here the argument frame of the verb needs to be constructed and wherever selectional preference dictates a particular sense that sense should be picked up . So in the 1st case, the context of serves has the object dinner which is edible , so serves in the sense of give/offer is chosen whereas in the second case serves has the object sector , so serves in the sense of running a service is chosen.

Example of Verb Argument frame

I gave a book to Rama

Verb: *Give*

Give

{

Agent:-<the give> animate

Direct object:-<the thing given> inanimate

Indirect object <beneficiary> animate/organisation

}

The semantic roles now can be marked in the words in the sentence:*-$[I]_{agent}$ gave a $[book]_{dobj}$to $[Rama]_{idobj}$*

Another example of Argument frame (Adjective)

I am fond of X.

Fond

{

Arg1 :- Prepositional Phrase (PP)

{

PP:- (of NP)

{

N:- somebody/something

}

}

}

4.2.1.2 Overlap Based Approaches

Overlap Based Approaches fall under knowledge based approach. The steps for Overlap Based approaches are the following :-

1. Machine readable dictionary is required to serve as sense repository.
2. It is required to find the overlap between features of an ambiguous word (sense bag) and the features of the words in its context (context bag).
3. These features could be sense definitions , example sentences , the gloss of the words, hypernymy , hyponymy , the gloss of the words in hypernymy , hyponymy, example sentences of hypernymy , hyponymy etc.
4. The features could also be given weights
5. The sense which has the maximum number of overlap is selected as the contextually appropriate sense.

4.2.1.2.1 Lesk Algorithm

This is based on the overlap idea. It contains a Sense bag which contains the words in the definition of a candidate sense of the ambiguous word and a Context bag which contains the words in the definition of each sense of the context word[14].

Example :- On burning coal we get ***ash.***

The noun ash has 3 senses from WordNet. These are:-

1. residue that remains when something is burned.
2. ash , ash tree (any of the various deciduous pinnate-leaved ornamental or timber
   trees of the genus Fraxinus)
3. ash (strong elastic wood of any of the various ash trees. Used for furniture, tool
   handles and sporting goods such as base ball bats)

The verb ash has 1 sense from WordNet

ash (convert into ashes)

Looking at the first three senses we observe that apart from the 1st sense of ash as

noun, no other sense has anything in common with the context in which *ash* appears in the example cited above . Here *burned* appears in the 1st sense and burning is in the context of ash in the given example. **So the 1st sense of ash is the winner sense.**
Disadvantages of Lesk Algorithm:- There are quite a few disadvantages of Lesk algorithm. These are:-

- Proper nouns can act as strong disambiguators but proper nouns are not present in Thesaurus.
- The algorithm considers the definition of senses at the surface level only but strong clues for disambiguation can come from deep analysis of senses when surface level analysis fails. Hence the accuracy level of Lesk algorithm is less (50% when tested on highly polysemous English words)

4.2.1.2.2Walker Algorithm

Walker Algorithm is a Thesaurus Based approach. It also falls under knowledge based approach. The steps are as follows[15]:-

Step 1: For each sense of the target word find the thesaurus category to which that sense belongs.

Step 2: Calculate the score for each sense by using the context words. A context words will add 1 to the score of the sense if the thesaurus category of the word matches that of the sense.

E.g. The money in this **bank** fetches an interest of 8% per annum

Here the Target word is bank

Clue words from the context: money, interest, annum, fetch

Table 4.1: Assigning scores in Walker's algorithm

	Sense 1: finance	Sense 2 : location
Money	+1	0
Interest	+1	0
Annum	+1	0
Fetch	0	0
Total	3	0

So we observe from the Table 4.1 that Sense 1:finance is the winner sense as it has the highest number i.e. 3

But the question is how would the machine know that, for e.g. money, interest , annum are all in the finance sense and not in the location sense. Therefore Walker's algorithm heavily depends on ontological information of words- their concept , meaning , hierarchy etc. Overall it can be said that the algorithm requires a costly ontological resource.

4.2.1.2.3 WSD using Conceptual Density

Here, a sense is selected based on the relatedness of the word-sense to the context. Relatedness is measured in terms of conceptual distance (i.e. how close the concept represented by the word and the concept represented by its context words are- it is actually obtained from the hpernymy-hyponymy relations of the WordNet.). This algorithm[15] is underpinned by the axiom that smaller the conceptual distance, higher will be the conceptual density (i.e. if all words in the context are strong indicators of a particular concept, then that concept will have a higher density). This method[15] fails to capture the strong clues provided by proper nouns in the context of the target word.

4.2.1.3 Knowledge Based Approaches-Comparison

The Table 4.2 shows the comparisons in as much as the efficiency of various knowledge based(KB) approaches.

Table 4.2 showing comparisons among KB approaches

Algorithm	Accuracy
WSD using Selectional Restrictions	44% on Brown Corpus
Lesk's algorithm	50-60% on short samples of "Pride and Prejudice" and some "news stories".
WSD using conceptual density	54% on Brown corpus.
Walker's algorithm	50% when tested on 10 highly polysemous English words.

4.2.2 Machine Learning Based Approaches

Machine Learning Based approaches are as follows:-

1. Supervised Approaches
2. Semi-supervised Approaches
3. Unsupervised Approaches

4.2.2.1 Supervised Approaches

Supervised methods need machine learning techniques and training corpora. Here the senses of a word for a new sentence are obtained from whatever we have learnt through the training corpora. The words around the target word are also called features. These features are learnt along with their weightage in the environment of the word to be disambiguated. This learning is used to disambiguate senses of words in a new sentence when it comes up.

The linguistic features used in training WSD can be roughly divided into two classes: collocational features and co-occurrence features. Collocational features encode information about the lexical inhabitants of specific positions located to the left or right of target word. Typical features include the word , the root form of the word and the word's part of speech etc.

For e.g. "I went to the **bank** to withdraw some money" where we need to disambiguate the word bank.

Collocation vector for the above e.g. is <I ,PP ,went, VBD, withdraw , VB , money, NN>

The co-occurrence feature consists of data of neighbouring words ignoring their exact position. In this approach the words themselves or their roots serve as features. The value of the feature is the number of times the word occurs in a region surrounding the target word.

4.2.2.1.1 Naïve Bayes Algorithm

It is a Supervised method based algorithm. The formula used here is

$\hat{s} = \mathbf{argmax}_{s\ \varepsilon\ senses} \mathbf{Pr}(s|V_w)$where

'V_w' is a feature vector consisting of:

- POS of w
- Semantic & Syntactic features of w

- Collocation vector
- Co-occurrence vector

By Bayes' rule

$\Pr(s|v_w) = \Pr(s) * \Pr(v_w|s) / \Pr(v_w)$ ……..... 4.1

By independence assumption

$\Pr(V_w|s) = \Pr(V_w^1|s).\ \Pr(V_w^2|s).\ \Pr(V_w^3|s)$……$\Pr(V_w^n|s)$

$= \prod_{i=1}^{n} Pr(V_w^i|s)$ …….....4.2

Hence from equation 4.1 and 4.2 we get

$\hat{s}=argmax_{s\in senses}\Pr(s)*\prod_{i=1}^{n} Pr(V_w^i|s)$ …….....4.3

For e.g. let us try to disambiguate word *bank* with Naïve Bayes algorithm in the sentence *"I went to the **bank** to withdraw some money"*.

POS of bank= Noun

SyntacticandSemantic features of bank could be <Noun ,Organization , place>

Collocation vector is <I ,went, withdraw , money>

There are mainly 2 senses of bank as Noun namely financial sense and river bank sense From equation 4.2 we see that $\Pr(V_w^2|s)$ which can be $\Pr(\text{Organization}|bank_{financial})$ will tilt the balance in favour of bank in the sense of a financial organization.

4.2.2.1.2 Decision List Algorithm

This is an algorithm predicated on the principle of nearby words provide strong and consistent clues as to the sense of a target word [17]. The steps of the algorithm are as follows:-

- Collect a large set of collocations for the ambiguous word.
- Calculate word-sense probability distributions for all such collocations.
- Calculate the log-likelihood ratio

$$Log\left(\frac{\Pr(Sense-A|Collocation_i}{\Pr(Sense-B|Collocation_i}\right)$$

- Higher log-likelihood = more predictive evidence.
- Collocations are ordered in a decision list, with most predictive collocations ranked highest.

To explicate the above approach let us take the sentence [17]

…plucking **flowers** affects ***plant growth***…

Training Data

Sense	Training Examples (Keyword in Context)
A	used to strain microscopic *plant* life from the ...
A	... zonal distribution of *plant* life . ...
A	close-up studies of *plant* life and natural ...
A	too rapid growth of aquatic *plant* life in water ...
A	... the proliferation of *plant* and animal life ...
A	establishment phase of the *plant* virus life cycle ...
A	... ...
B	... ...
B	computer **manufacturing** *plant* and adjacent ...
B	discovered at a St. Louis *plant* **manufacturing**
B	... copper **manufacturing** *plant* found that they
B	copper wire **manufacturing** *plant* , for example ...
B	's cement **manufacturing** *plant* in Alpena ...
B	polystyrene **manufacturing** *plant* at its Dow ...
B	company **manufacturing** *plant* is in Orlando ...

Resultant Decision List

LogL	Collocation	Sense
10.12	*plant* growth	⇒ A
9.68	car (within ±*k* words)	⇒ B
9.64	*plant* height	⇒ A
9.61	union (within ±*k* words)	⇒ B
9.54	equipment (within ±*k* words)	⇒ B
9.51	assembly *plant*	⇒ B
9.50	nuclear *plant*	⇒ B
9.31	flower (within ±*k* words)	⇒ A
9.24	job (within ±*k* words)	⇒ B
9.03	fruit (within ±*k* words)	⇒ A
9.02	*plant* species	⇒ A
...	...	

Fig 4.1 Example to explicate Decision List Algorithm

From the Fig 4.1 above we see that large set of collocations for the ambiguous word plant in the botanical sense and manufacturing sense is constructed and tagged with either sense A or sense B. These collocations are used to disambiguate a word in a new sentence.

4.2.2.1.3 Exemplar based WSD (K-NN)

An exemplar based classifier [21] is constructed for each word to be disambiguated**.** The steps in this algorithms are as follows:-

1. From each sense marked sentence containing the ambiguous word , a training example is constructed using:
    - POS of w as well as POS of neighboring words.
    - Local collocations
    - Co-occurrence vector
    - Morphological features
    - Subject-verb syntactic dependencies
2. Given a test sentence containing the ambiguous word, a test example is similarly constructed.

3. The test example is then compared to all training examples and the k-closest training examples are selected.
4. The sense which is most prevalent amongst these "k" examples is then selected as the correct sense.

4.2.2.1.4 WSD Using Support Vector Machine

SVM is a binary classifier which finds a hyperplane with the largest margin that separates training examples into two classes [32].

- As SVMs are binary classifiers, a separate classifier is built for each sense of the word
- Training Phase: Using a tagged corpus, for every sense of the word a SVM is trained using the following features:
    - POS of target word w as well as POS of neighbouring words.
    - Local collocations
    - Co-occurrence vector
    - Features based on syntactic relations (e.g. headword, POS of headword, voice of head word etc.)
- Testing Phase: Given a test sentence, a test example is constructed using the above features and fed as input to each binary classifier.

The correct sense is selected based on the label returned by each classifier.

4.2.2.1.5 Supervised Approaches – Comparison of methodology

General Comments

- Use corpus evidence instead of relying of dictionary defined senses.
- Can capture important clues provided by proper nouns because proper nouns do appear in a corpus.

Naïve Bayes

- Suffers from data sparseness.
- Since the scores are a product of probabilities, some weak features might pull down the overall score for a sense.
- A large number of parameters need to be trained.

Decision Lists

- A word-specific classifier. A separate classifier needs to be trained for each word.
- Uses the single most predictive feature which eliminates the drawback of Naïve Bayes.

Exemplar Based K-NN

- A word-specific classifier.
- Will not work for unknown words which do not appear in the corpus.
- Uses a diverse set of features (including morphological and noun-subject-verb pairs)

SVM

- A word-sense specific classifier.
- Gives the highest improvement over the baseline accuracy.
- Uses a diverse set of features.

4.2.2.1.6 Supervised Approaches- Comparison of Performance

Table 4.3 shows the comparison in terms of performance among various Supervised approaches.

Table 4.3 : Performance comparison among Supervised Approaches

Approach	Average Precision	Average Recall	Corpus	Average Base line Accuracy
Naïve Bayes	64.13%	Not reported	Senseval3 – All Words Task	60.90%
Decision List	96%	Not reported	Tested on a set of 12 highly polysemous English words	63.9%
Exemplar based WSD (K-NN)	68.6%	Not reported	WSJ6 containing 191 content words	63.7%
SVM	72.4%	72.4%	Senseval 3 – Lexical sample task (Used for disambiguation of 57 words)	55.2

4.2.2.2 Semi Supervised Approach

The basic ideas behind Semi Supervised WSD are as follows:-

1. Need seed training data.
2. Train using seed data
3. Tag unseen data from the corpus
4. Manually correct the tags that were generated wrongly in step 3.
5. Retrain the model using larger data
6. Repeat steps 3,4 until satisfactory accuracy level is obtained.

4.2.2.2.1 Semi Supervised Decision List Algorithm

This algorithm is based on Yarowsky's supervised algorithm [17] that uses Decision Lists. The key steps of the algorithm [18] are:-

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

This algorithm exploits two important properties of human language [18]. These are:-

- One sense per collocation- Nearby words provide strong and consistent clues to the sense of a target word. There is a strong tendency for words to exhibit only one sense in a given collocation . This effect varies depending on the type of collocation. It is strongest for immediately adjacent collocations, and weakens with distance. It is much stronger for words in a predicate-argument relationship than for arbitrary associations at equivalent distance. It is very much stronger for collocations with content words than those with function words.
- One sense per discourse- The sense of a target word is highly consistent within any given document. Words strongly tend to exhibit only one sense in a given discourse or document.

4.2.2.3 Unsupervised Approach

Unsupervised learning is the greatest challenge for WSD researchers. The underlying assumption is that similar senses occur in similar contexts, and thus senses can be induced from text by clustering word occurrences using some measure of similarity of context,a task referred to as word sense induction or discrimination. Then, new occurrences of the word can be classified into the closest induced clusters/senses. The key question in Unsupervised WSD is how to put the sense label on the word when the training data did not have any label. The solution is lexicon of labels which are not used to mark the training data ; they are rather used to generate the labels once we have determined the sense of the word. Performance for Unsupervised methods have been empirically seen to be lower than the Supervised methods described above.

4.2.2.3.1 Hyperlex

This is a graph-based Unsupervised WSD approach proposed byVeronis, [29]. 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 five 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 [29] stated that the graph thus created has the properties of "small worlds" "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. For 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 two.

Detecting Root Hubs

1. The steps for detecting root hubs are as follows:-
2. Construct co-occurrence graph, G.
3. Arrange nodes in G in decreasing order of in-degree.
4. Select the node from G which has the highest frequency. This node will be the hub of the first high density component.
5. Delete this hub and all its neighbours from G.
6. Repeat Step 3 and 4 to detect the hubs of other high density components

4.2.2.3.2 Yarowosky's Algorithm (WSD Using Roget's Thesaurus Categories)

This algorithm is based on the following three observations [16]:-

- Different conceptual classes of words (say ANIMALS and MACHINES) tend to appear in recognizably different contexts.
- Different word senses belong to different conceptual classes (E.g. crane).
- A context based discriminator for the conceptual classes can serve as a context based discriminator for the members of those classes.

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

$$ARGMAX_{Rcat} \sum_{wincontext} \log \left(\frac{\Pr(w|Rcat)*\Pr(Rcat)}{\Pr(w)}\right)$$

The table 4.4 shows the implementation of Yarowsky's algorithm on the targetword crane. A crane might mean a machine operated for construction purpose (Roget'scategory of TOOLS/MACHINE) or a bird (Roget's category of ANIMAL/INSECT). Byfinding the context words for word crane and finding how much weight (similarity) theyimpose on each sense of crane, the winner sense is selected.

Table 4.4Example list showing a run of Yarowsky's algorithm

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

4.2.2.3.3 Unsupervised WSD using Parallel Corpora

This approach [28] 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. This 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.

4.2.2.3.4 Dekang Lin's approach

The algorithm proposed by Lin [22] uses syntactic dependency as local context to resolve word sense disambiguity. Most corpus-based WSD algorithms determine the meanings of polysemous words by exploiting their local contexts. A basic intuition that underpinned those algorithms is the following:

Two occurrences of the same word have identical meanings if they have similar local contexts.

But Lin's algorithm [22] is underpinned by a different intuition:-

Two different words are likely to have similar meanings if they occur in identical local contexts.

The algorithm [22] does not require a sense-tagged corpus and exploits the fact that two different words are likely to have similar meanings if they occur in identical local contexts. Here similarity is directly proportional to the probability that the two words have the same super class (Hypernym) .

For e.g. let us take the sentence:-

The new facility will employ 500 of the existing 600 employees

The word "facility" has 5 possible meanings in WordNet :

- installation
- proficiency/technique
- adeptness
- readiness
- toilet/bathroom

The words in the table 4.4 are the subjects of “employ” in a 25-million-word Wall Street Journal corpus.

Table 4.5 Subjects of "employ" in a 25-million-word Wall Street Journal corpus [22].

Word	Freq	Log Likelihood
ORG	64	50.4
Plant	14	31.0
Company	27	28.6
Industry	9	14.6
Unit	9	9.32
Aerospace	2	5.81
Memory device	1	5.79
Pilot	2	5.37

The "freq" column in table 4.4 are the number of times the words in the “word”column occurred as the subject of "employ".The meaning of "facility" in the above sentence can be determined by choosing one of its 5 senses that is most similar to the meanings of words in Table 4.5. Through this way, a polysemous word is disambiguated with past usages of other words which appear in the same local context as the polysemous word to be disambiguated. Whether or not the polysemous or the target word itself appears in the corpus is irrelevant.

4.2.2.3.5 Unsupervised Approaches – Comparison of Performance

The table 4.6 shows the comparison of performances among various unsupervised approaches.

Table 4.6 Comparison among various Unsupervised Approaches

Approach	Precision	Average Recall	Corpus	Baseline
Lin's Algorithm	68.5%. The result was considered to be correct if the similarity between the predicted sense and actual sense was greater than 0.27	Not reported	Trained using WSJ corpus containing 25 million words. Tested on 7 SemCor files containing 2832 polysemous nouns	64.2%
Hyperlex	97%	82% (words which were not tagged with confidence>threshold were left untagged)	Tested on a set of 10 highly polysemous French words	73%
WSD using Roget's Thesaurus categories	92% (average degree of polysemy was 3)	92% (average degree of polysemy was 3)	Tested on a set of 12 highly polysemous English words	Not reported
WSD using parallel corpora	SM: 62.4% CM: 67.2%	SM: 61.6% CM: 65.1%	Trained using a English Spanish parallel corpus Tested using Senseval 2 – AllWords task (only nouns were considered)	Not reported

4.2.2.3.6 Unsupervised Approach - Comparison of methodology

Unsupervised approaches combine the advantages of supervised and knowledge based approaches. They resemble supervised approaches in that they extract evidence from corpus and they resemble knowledge based approaches in that they do not need tagged corpus. A summary of the key algorithms implementing Unsupervised approaches is given below:-

Lin's Algorithm

- A general purpose broad coverage approach.
- Can even work for words which do not appear in the corpus.

Hyperlex

- Use of small world properties was a first of its kind approach for automatically extracting corpus evidence.
- A word-specific classifier.
- The algorithm would fail to distinguish between finer senses of a word (e.g. the medicinal and narcotic senses of "drug")

Yarowsky's Algorithm(WSD using Roget's thesaurus categories)

- A broad coverage classifier.
- Can be used for words which do not appear in the corpus. But it was not tested on an "all word corpus".

WSD using Parallel Corpora

- Can distinguish even between finer senses of a word because even finer senses of a word get translated as distinct words.
- Needs a word aligned parallel corpora which is difficult to get.
- An exceptionally large number of parameters need to be trained.

4.2.3 Hybrid Approach

This sort of approach acquires knowledge from diverse sources like WordNet and corpus. It is similar to Supervised approach in the sense that some sense tagging is required here. It is similar to Unsupervised approach in the sense that information in the form of corpus evidence is utilised here. On the whole it can combine information from multiple knowledge sources and use a comparatively small amount of tagged data.We describe some algorithms encapsulating the hybrid approach below:-

4.2.3.1 Sense Learner

This algorithm [30] uses some tagged data to build a semantic language model for words seen in the training corpus. It also uses WordNet to derive semantic generalizations for words which are not observed or seen in the corpus.

The basic steps of this algorithm are as follows:-

- For each POS tag, using the corpus, a training set is constructed.
- Each training example is represented as a feature vector and a class label which is word#sense
- In the testing phase, for each test sentence, a similar feature vector is constructed.
- The trained classifier is used to predict the word and the sense.

The algorithm improvises on Lin's approach by exploiting semantic dependencies through the relations in WordNet.

For e.g. let us consider the following sentencefrom SemCor: "The Fulton County Grand Jurysaid Friday an investigation of Atlanta's recentprimary election produced "no evidence" that anyirregularities took place" [30].

After parsing by a dependency parser, a verb-object link/relation could be inferred from the words "produced" and "evidence". The hypernymy tree is looked up for each pair involved in syntactic dependency and a feature vector is created as follows:-

<produce ,expose , show , evidence, information, cognition, psychological_features>

Now if the test data is "expose meaningful information." [30], we identifyan object-verb relation between expose and information.Although none of the words in the pair"expose information" appear in the training corpus,by looking up the IS-A hierarchy from WordNet,we will be able to successfully disambiguatethis pair, as both "expose" and "information" appearin the feature vector constructed above.

4.2.3.2 An Iterative Approach to Word Sense Disambiguation

Following are the key steps in this approach[26]:-

- It uses semantic relations (synonymy and hypernymy) form WordNet.
- It extracts collocational and contextual information form WordNet (gloss) and a small amount of tagged data.
- Monosemic words in the context serve as a seed set of disambiguated words.

- In each iteration new words are disambiguated based on their semantic distance from already disambiguated words.

4.2.3.3 Structural Semantic Interconnections(SSI)

This approach [36] is an iterative approach and it uses the following relations:-

- hypernymy (car#1 is a kind of vehicle#1) denoted by (kind-of)
- hyponymy (the inverse of hypernymy) denoted by (has-kind)
- meronymy (room#1 has-part wall#1) denoted by (has-part)
- holonymy (the inverse of meronymy) denoted by (part-of)
- pertainymy (dental#1 pertains-to tooth#1) denoted by (pert)
- attribute (dry#1 value-of wetness#1) denoted by (attr)
- similarity (beautiful#1 similar-to pretty#1) denoted by (sim)
- gloss denoted by (gloss)
- context denoted by (context)
- domain denoted by (dl)

Monosemic words serve as the seed set for disambiguation in SSI[36]. SSI builds a semantic relation graph for the multiple senses of a word that needs to be disambiguated.

4.2.3.4 Comparison of Performance among the various Hybrid approaches

The table 4.7 shows Comparison of Performance among the various Hybrid approaches

Table 4.7 Comparison of Performance among the various Hybrid approaches

Approach	Precision	Average Recall	Corpus	Baseline
An Iterative Approach to WSD	92%	55%	Trained using 179 texts from SemCor. Tested using 52 texts created from 6 SemCor files	Not reported
Sense Learner	64.6	64.6	SenseEval-3 All Words Task	60.9%
SSI	68.5	68.4	SenseEval-3 Gloss Disambiguation Task	Not reported

4.3 Performance Metric of a WSD system

The Performances Metrics for a WSD system, as with some other NLP systems, are:-

Precision (P%) :- It is defined as the measure of the selected items that the system got right.

Precision(P%) = $\frac{t_p}{t_p + f_p}$ where t_p and f_p are true positive and false positive respectively.

Recall (R%):- It is defined as the proportion of target items that the system selected.

Recall $= \frac{t_p}{t_p + f_n}$ where f_n is false negative.

In NLP applications we can generally trade-off Precision and Recall (one can select all the documents in the collection and get 100% Recall but very poor Precision. Similarly one can select very small number of documents and get a high Precision but very poor recall) . For this reason it is convenient to combine both Precision and Recall into a single measure of overall performance which we call F measure.

F% $= \frac{2PR}{P+R}$

4.4 Chapter Summary

This chapter has discussed WSD and its various approaches in significant details. The algorithms encapsulating the various approaches have also been discussed and their performances on different types of Corpus have been enumerated. The pros and cons of various approaches to WSD and related algorithms have been studied in detail so as to form a well rounded view about the applicability and implementational aspect of WSD. The performance metrics of WSD system have also been discussed in this chapter.