Utilizing Corpus Statistics for Hindi Word Sense Disambiguation

Satyendr Singh and Tanveer Siddiqui
Department of Electronics and Communication, University of Allahabad, India

Abstract: *Word Sense Disambiguation (WSD) is the task of computational assignment of correct sense of a polysemous word in a given context. This paper compares three WSD algorithms for Hindi WSD based on corpus statistics. The first algorithm, called corpus-based lesk, uses sense definitions and a sense tagged training corpus to learn weights of Content Words (CWs). These weights are used in the disambiguation process to assign a score to each sense. We experimented with four metrics for computing weight of matching words Term Frequency (TF), Inverse Document Frequency (IDF), Term Frequency-Inverse Document frequency (TF-IDF) and CW in a fixed window size. The second algorithm uses conditional probability of words and phrases co-occurring with each sense of an ambiguous word in disambiguation. The third algorithm is based on the classification information model. The first method yields an overall maximum precision of 85.87% using TF-IDF weighting scheme. The WSD algorithm using word co-occurrence statistics results in an average precision of 68.73%. The WSD algorithm using classification information model results in an average precision of 76.34%. All the three algorithms perform significantly better than direct overlap method in which case we achieve an average precision of 47.87%.*

Keywords: *Supervised hindi WSD, corpus based lesk, TF-IDF, statistical WSD, word co-occurrence, information theory, classification information model.*

Received August 15, 2013; accepted May 6, 2014; published online September 15, 2015

1. Introduction

Natural languages contain words bearing multiple meaning and Hindi is not an exception. Human beings can easily arrive at the correct sense (meaning) of a word using the context in which it is used. However, the dependency between meaning and context is not well understood and hence computational representation of context is difficult. This makes automatic identification of correct sense of a word in a given context a difficult task. This task is referred to as Word Sense Disambiguation (WSD) and is a central research topic in Natural Language Processing (NLP). WSD is characterized as an intermediate task in many NLP applications and is essential in applications requiring broad coverage language understanding, e.g., machine translation, text summarization, question answering, etc.

As mentioned earlier, identifying correct sense of a word requires consideration of the context. In NLP, definition of context is closely related to specific task, domain and application. Most of the WSD techniques consider context as the text surrounding an ambiguous word, usually in a fixed size window keeping ambiguous word in the middle. This context is utilized in a variety of ways. The simplest is to consider the number of matching words between dictionary definition of words appearing in a test instance and the dictionary definitions of various senses of the word being disambiguated. However, this direct overlap method could recognize similarity only when an exact match occurs. One way to overcome this limitation is to extend the context being matched so, as to increase the chances of matching words. If a sense tagged corpus is available this can be done by considering training instances of various senses as extended sense definitions. Other ways include extension of matching context with the help of semantic relations like synonym, hypernym, etc., combining local context with semantic similarity [12], utilizing statistical information gathered over some corpus, etc. The availability of sense tagged corpora has contributed a lot to the recent advances in WSD. Most accurate WSD systems use some supervised learning algorithm to learn contextual rules or classification models automatically from sense-annotated examples. Several supervised approaches including Naïve Bayes [11], k-NN [17] and Support Vector Machine (SVM) classifiers [15] have shown high accuracy in WSD. The majority of work on WSD is focused on English and other European languages and standard test corpora are available for these languages. The lack of such standards put a major hindrance on WSD research for Hindi and other Indian languages.

In this paper, we propose three WSD algorithms for Hindi which use statistical features extracted from a sense tagged corpus for disambiguation. The first algorithm is an extension of basic lesk algorithm. It considers sense tagged training corpus as additional information source and uses it along with dictionary definition of senses for disambiguation. Instead of measuring overlap in terms of matching words, we

make use of Term Frequency-Inverse Document Frequency (TF-IDF) weighting schemes commonly used in information retrieval for computing weights of matching Content Words (CWs). The hypothesis is that the chances of a word occurring in context of a particular sense will be high as compared to other senses. For example the target word सोना (sona) has 2 senses: Sleep and gold. The chances of occurrence of the word चांदी (chandi/silver) in the context of सोना in gold sense is high as compared to सोना in sleep sense. The high frequency value in a particular sense will improve the chances of correct disambiguation. The second algorithm uses conditional probability of co-occurring words and phrases collected over a sense tagged training corpus for disambiguation. The underlying assumption is that words and phrases co-occurring with a particular sense of an ambiguous word are good indicators of its sense and hence the co-occurrence information is expected to contribute positively in sense identification. The third algorithm is based on the classification information model. It uses the classification information of surrounding words of target polysemous word for performing sense identification.

The rest of the paper is organized as follows: Related work is reviewed in section 2. Section 3 discusses WSD Algorithms used in this work. The details of the data set and experiments conducted are provided in section 4. Section 5 provides results and discussion and finally conclusions are drawn in section 6.

2. Related Works

There are two broad categories of existing WSD techniques: Knowledge-based and corpus-based. Knowledge based approaches rely on the availability of lexicon, thesaurus or dictionary for performing disambiguation. Corpus-based approaches use sense tagged corpus (supervised) or raw corpus (unsupervised) for performing disambiguation.

lesk [16] was one of the pioneer works in knowledge-based WSD. He performed contextual overlapping between sense definitions of words occurring in neighbourhood of ambiguous word and dictionary definitions of each sense for disambiguation. The sense maximizing the score was selected as the winner sense. Following lesk several variants of lesk have been proposed [1, 2, 3, 24]. Banerjee and Pederson [2] proposed adapted lesk algorithm which uses WordNet as dictionary. They explored and used various semantic relations such as hypernym, hyponym, meronym, tryponym and attribute of each word glosses in disambiguation. In another work, Banerjee and Pederson [3] proposed a new measure of semantic relatedness between concepts. This measure was based on the number of overlaps in glosses. They included the glosses of other concepts to which a concept is related in the WordNet concept hierarchy. Vasilescu *et al*. [24] performed comparative evaluation of variants of lesk's algorithm. They found performance of simplified lesk algorithm better than the original lesk's algorithm. Baldwin *et al*. [1] proposed a new method of MRD-based WSD using definition expansion via ontology. Their work was build on the work of lesk [16] and Banerjee and Pederson [3]. They experimented with character and word-based tokenization, definition extension being based on the words in original definition sentences. They also experimented with a range of lexical relations including both sense-sensitive and sense-insensitive expansion. Evaluation was done on two Japanese datasets: example sentences from the Hinoki Sensebank and a retagged version of the Senseval-2 Japanese dictionary task. Kavitha [6] proposed three semantic similarity measures for measuring similarity between words and sentences. The first method was based on association rule mining for computing similarity. The second method utilized SVM classifier and integrated page counts and snippets returned by web search engine for computing similarity. The third method was based on sequential clustering algorithm.

Sinha *et al*. [21] used an extension of lesk based approach for Hindi WSD. They performed contextual overlapping between sentential context and extended sense definitions. Extended sense definitions included words extracted from synonyms, glosses, example sentences, hypernyms, glosses of hypernyms, example sentences of hypernyms, hyponyms, glosses of hypernyms, example sentences of hypernyms, meronyms, glosses of meronyms, example sentences of meronyms. Context bag was created by extracting words in the neighbourhood of target word. Winner sense was assigned to one which maximized the overlap. Singh and Siddiqui [19] evaluated the effect of stemming, stop word removal and context window size for Hindi WSD in a lesk like setting. They reported improvement of 9.24% in precision after stemming and stop word removal over the baseline. Singh *et al.* [20] adapted and evaluated leacock chodorow semantic relatedness measure for Hindi WSD. Leacock chodorow measure uses Hindi WordNet hierarchy to learn semantics of words and is based on the length of noun concepts in an is-a hierarchy. Evaluation was done on 20 Hindi polysemous nouns and they achieved an accuracy of 60.65% using this measure. Khapra *et al.* [7] studied domain specific WSD for nouns, adjectives and adverbs for English, Hindi and Marathi. They used dominant senses of words in specific domains for performing disambiguation. An accuracy of 65% on F1-score was reported for all the three languages. Khapra *et al.* [9] projected WordNet and corpus parameters in a multilingual setting involving Hindi, Marathi, Bengali and Tamil. Their method was based on a novel synset based multilingual dictionary and the observation that within a domain the distribution of senses remains more or less invariant across languages. They projected parameters from Hindi to other three languages using two different WSD algorithms. Evaluation was performed on tourism and health domains and they achieved F1-score of 75% for

three languages. Khapra *et al.* [8] performed bilingual bootstrapping between two resource deprived languages, both having a small amount of seed annotated data and a large amount of untagged data. They trained a model using the seed annotated data of one language to annotate the untagged data of another language and vice versa using parameter projection. They evaluated bilingual bootstrapping algorithm on two different domains with small seed sizes using Hindi and Marathi as the language pair.

In SENSEVAL, held in 1998 for English language WSD task, corpus based lesk was used as baseline [10]. Inverse Document Frequency (IDF) was used for computing the weights of matching words in dictionary definitions and training corpus.

In this work, we use TF-IDF weighting scheme for Hindi WSD task which is widely used in information retrieval. Earlier evidences of using word co-occurrence statistics in WSD research include [22, 23, 25]. Yang *et al.* [25] introduced a WSD method based on sememe co-occurrence frequency. They used Hownet as information source and a database of co-occurrence frequency of sememes was used in performing WSD. Turney [23] described the National Research Council WSD system. The system was supervised and uses weka machine learning software and Brill's rule-based part-of-speech tagger. They represented headwords as feature vectors, which were both syntactic and semantic. They generated semantic features using word co-occurrence probabilities. We use word co-occurrence statistics for Hindi WSD. Suderman [22] developed and tested a supervised WSD system named wisdom that used co-occurrence statistics in a small window for performing disambiguation.

The work on WSD based on information theory is reported in [13, 14]. Lee *et al.* [13] used the classification information of surrounding words of target polysemous word for predicting the correct sense of target word. Their classification information model was based on the Shannon's information theory. They achieved an accuracy of 84.6% for the Korean dataset and 80.0% for the English dataset. Lee *et al.* [14] used a classification information model for performing disambiguation. They extracted classification information from training instances and used weighted sum of whole individual decisions derived from features contained in the instance for performing sense disambiguation. In this work we use classification information model for Hindi WSD.

3. Algorithmic Formulation

3.1. Corpus Based Lesk Algorithm

Corpus based lesk algorithm is a supervised WSD algorithm that uses sense definitions and sense tagged training corpus for performing disambiguation. Sense definitions and sense tagged training corpus is used for computing weights of matching CWs in context of target word. The context of target word is taken as a list of words which appears in $a \pm n$ window size, with the target word in the middle. For a window size of n, context vector size is $2n+1$. Firstly, stop words are removed from sense definitions, sense tagged training corpus and test corpus. A context vector is formed from the test corpus with the target word in the middle. The target word in the context vector is dropped and duplicates are removed.

A sample test instance for target word सोना (sona) is shown in Figure 1:

सोमवार की तुलना में मंगलवार को उदयपुर सर्राफा में सोना प्रति दस ग्राम के भाव 200 रुपए कम हुए तो चांदी में पांच सौ रुपए किलो की गिरावट रही।

{Somvar ki tulna me mangalwar ko Udaipur sarafa mein sona prati das gram ke bhaw do sau rupay kam hue to chandi me paach sau rupay kilo ki girawat rahi.}

(As compared to Monday, on Tuesday in Udaipur bullion market, the price of gold per ten grams dropped by Rupees two hundred and the price of silver dropped by Rupees five hundred per kilogram.)

Figure 1. A sample test instance for target word सोना (sona).

The context vector for target word सोना (sona) for window size 5 after dropping target word and removing duplicates is:

[सोमवार, तुलना, मंगलवार, उदयपुर, सर्राफा, प्रति, दस, ग्राम, भाव, रुपए]

The words in this context vector are searched in sense definitions and sense tagged training corpus. If a match is found the weight of the word is computed. In this work extended sense definitions are used which comprise of synsets, glosses and example sentence of target polysemous word. The extended sense definition is treated as an instance for computing weight. Using the sum of weights of matching CWs in context vector a score is assigned to each sense. The sense having maximum weight is the winner sense. We experimented with 4 different weighting schemes:

- Term Frequency (TF) of a Word: Is simply the number of occurrences of a word in a document. *TF* is normalized by dividing it by the maximum frequency of word in that document. Normalized *TF* is computed across a single document (comprising of sense definition and sense tagged training corpus for a particular sense).

$$TF_{(t,\,d)} = TF_{(t,\,d)} / TF_{max\,(d)}$$

- *IDF* of a Word *t*: Is defined as log of the ratio of the total number of documents and the number of documents containing the word *t*. For obtaining *IDF*, all the sense definitions and sense tagged training corpus for all the senses of a target word are merged and *IDF* is computed across it.

$$IDF_t = log(N/N_t)$$

- *TF-IDF*: Is simply the product of *TF* and *IDF* of a word.

$$TF\text{-}IDF = TF_{(t,\,d)} * IDF_t$$

- CW in a Fixed Window Size: The CW in a fixed window size is computed as the ratio of total number of instances containing the word in a window to total number of instances. The window size is same as the size of context vector.

The steps in corpus based lesk algorithm are summarized below:

1. Remove stop words from sense definitions, sense tagged training corpus and test corpus.
2. For each sense s of a polysemous word.
3. score(s)←0.
4. Identify set of unique words W in surrounding context window of test instances.
5. For each word w in W.
   For each sense s
   If w occurs in the sense definitions or sense tagged training corpus of s.
   score(s) ←score(s)+score(w)
6. Choose sense with maximum score(s).

3.2. WSD Algorithm Using Word Co-Occurrence

This algorithm attempts to disambiguate a word on the basis of co-occurrence words and phrases. The co-occurrence statistics is collected over a sense tagged training corpus. We have used a fixed size window surrounding an ambiguous word for extracting co-occurrence words and phrases. In this work, we take two words to the right and two words to the left of an ambiguous word w. For each word appearing in a±2 window of w, we keep a count of number of times the word appears with each senses of w. For example, for the target word सोना (sona) in the context as shown in Figure 1.

The words adjacent to सोना (sona) are सर्राफा, में, प्रति, दस, these constitute the co-occurrence words. Similarly, we extract co-occurring phrases in the window. For extracting phrases we remove the target word from the window and extract consecutive pair of adjacent words as phrases. In addition, we consider one more phrase consisting of all the words as co-occurring phrases. For example, if a word appears in the phrase "*a b w c d*" then the strings *ab*, *bc*, *cd* and *abcd* are referred to as co-occurring phrases.

For the example in Figure 1, relevant phrases are सर्राफा में, में प्रति, प्रति दस and सर्राफा में प्रति दस.

For each of these phrases, the co-occurrence frequency with each sense of w is computed. The co-occurrence count is then converted into conditional probability as follows:

$$P(c_i) = n_i / \sum_{i=1}^{n} n_j$$

Here, n_i is number of times co-occurring word (or phrase) c has appeared with i^{th} sense of w. n is the number of possible senses of word w.

During disambiguation stage, words and phrases occurring in the test instances are extracted. The test vector is matched with the co-occurring words and phrases of each sense of the target word w. A score is assigned to each sense by adding conditional probabilities of matching words and phrases. The sense which maximizes the score is assumed to be the winner sense.

3.3. WSD Algorithm Using Classification Information Model

Classification information model is based on the Shannon's information theory. This model classifies the input instance by the binary features representing the instance. It uses the classification information of surrounding words for performing disambiguation. The classification information of surrounding word consists of Most Probable Class (MPC) and Discrimination Score (DS). The MPC of a word represents the most closely related sense of the target word. The DS represents the degree of correlation between the surrounding word and most probable class.

Shannon used the concept of entropy for measuring the uncertainty in a message. The entropy becomes the average information value for a given message. The entropy H, average information value of n messages is computed as given in Equation 1:

$$H = -\sum_{i=1}^{n} p_j \log_2 p_j \quad (1)$$

Where p_j is the occurrence probability of the message.

According to this theory, the surrounding words of the target word can decrease the uncertainty of the target word. Surrounding words having more discriminating ability are ones which can decrease much uncertainty.

The noise produced by a surrounding word x_k can be computed using Equation 2 as:

$$noise_k = -\sum_{i=1}^{n} nor(p(x_k|sense_j)) \log_2 nor(p(x_k|sense_j)) \quad (2)$$

Where $nor(p(x_k|sense_j))$ is the normalized occurrence probability of surrounding word x_k in j^{th} sense of target polysemous word and n is the number of senses.

The normalized occurrence probability of x_k in sense i can be computed as the ratio of probability of x_k in i^{th} sense of target word to the sum of probabilities of x_k across all senses, as given in Equation 3:

$$nor(p(x_k|sense_i)) = p(x_k|sense_i) / \sum_{i=1}^{n} (p(x_k|sense_i)) \quad (3)$$

In this experiment we used normalized occurrence probability because according to Lee *et al*. [13] it prevents the model from overemphasizing the imbalance of size of training dataset among various senses.

The probability of x_k in i^{th} sense of target word can be computed as the ratio of frequency of x_k in i^{th} sense of target word to frequency of x_k across all senses, as given in Equation 4.

$$p(p(x_k|sense_i)) = frequency(x_k|sense_i) / frequency(x_k) \quad (4)$$

The words having higher noise has lower discrimination ability. The Discrimination Score (DS) of a word can be measured as an inverse function of noise and is given in Equation 5.

$$DS_k = log_2 n - noise_k \quad (5)$$

The normalized MPC of surrounding word x_k can be computed by the Equation 6.

$$MPC_k = argmax_i \, nor(p(x_k | sense_i)) \quad (6)$$

Firstly, we remove the stop words from the training and test instances as stop words provides less classification information in the sense decision. A context vector is formed from the test instances for a window size with the target word in the middle. The target word in the context vector is dropped. In this work, we have used a fixed window size of 4. The context vector consists of four words to the left and four words to the right of target word.

A sample test instance for target word **हल** (hal-ploughing instrument) is shown in Figure 2.

किसान यह सुनकर बहुत दुखी हुआ, लेकिन और कोई रास्ता भी नहीं था। वह खेत में पहुँचा और हल में एक ओर बैल को और दूसरी ओर अपनी पत्नी को जोतकर खेत जोतने लगा।

{Kisan yah sunkar bahut dukhi hua, lakin aur koi rasta bhi nahi tha. Wah khet me pahucha aur hal me ek oor bail ko aur doosri oor apni patni ko jotkar khet jotne laga}

(Hearing this farmer felt very sad but he had no other option. He went to the field and at one end of ploughing instrument he used a bull and at other end his wife and started ploughing the field.)

Figure 2. A sample test instance for target word **हल** (hal-ploughing instrument).

The context vector formed from the above test instance for window size of 4 is shown below:

[दुखी, रास्ता, खेत, पहुँचा, बैल, पत्नी, जोतकर, खेत]

The sense of the target word can be computed by the summation of DS of all the words in the context vector. The DS of all the words is computed across training instances using Equation 5. The MPC of word is computed across training instances using Equation 6. It is the sense in which the surrounding word has maximum normalized occurrence probability. The sense of the target word for a given context vector can be determined by the Equation 7.

$$MPC(S) = argmax_i \sum_{i=1}^{n} DS_k(i) \quad (7)$$

Where the *DS* of x_k over $sense_i$, $DS_k(i)$, is defined as:

$$DS_k(i) = DS_k \text{ if } i \text{ is the MPC of } x_k \text{ and } 0 \text{ otherwise} \quad (8)$$

The sense which maximizes *MPC* score as obtained by Equation 7 is assigned the winner sense. The sense decision of the above context vector is depicted in Table 1. The surroundings words, their MPC and *DS* are computed using Equations 5 and 6. If a surrounding word is not found in the training instances, then the *DS* of that word is 0.0 and *MPC* of that word is none.

Table 1. Sense disambiguation using the classification information model.

Surrounding Words	Training		Testing	
	MPC_k	DS_k	$DS_k(i)$	
			Sense1	Sense2
दुखी	none	0.0	0.0	0.0
रास्ता	1	1.0	1.0	0.0
खेत	2	1.0	0.0	1.0
पहुँचा	none	0.0	0.0	0.0
बैल	2	1.0	0.0	1.0
पत्नी	2	1.0	0.0	1.0
जोतकर	none	0.0	0.0	0.0
खेत	2	1.0	0.0	1.0
$\sum_{k=1}^{g} DS_k(i)$			1.0	4.0
Sense of Target Word			Sense 2	

4. Dataset and Experiments

4.1. Dataset

For evaluation of all the three algorithms, we have developed and used a sense annotated Hindi corpus [18] consisting of 60 polysemous nouns as shown in Table 2. The sense annotated Hindi corpus is available at Indian Language Technology Proliferation and Deployment Centre of Technology Development for Indian Languages (TDIL) portal. Test instances are collected from Hindi Corpus [4] created by Centre for Indian Language Technology (CFILT), Indian Institute of Technology (IIT) Bombay by firing queries derived from sense definitions. Test instances are also collected by performing search using Google and www.khoj.com. These instances are from varying domains including medical, news, stories, science, literature etc. The sense inventory is derived from Hindi WordNet [5]. Some of the senses having very fine grained sense distinctions are merged. For some of the senses which are not commonly used we could not find instances and hence we dropped them. The sense annotated Hindi corpus comprises a total of 7506 instances. The average number of instances per word is 125.1, average number of instances per sense is 49.70 and average number of senses per word is 2.51. For evaluating all these three algorithms, 70% instances of each sense of every target word have been used as training corpus and 30% as testing corpus. For keeping the test and training corpus incoherent, we have picked top 70% instances of each sense for training purpose and bottom 30% of instances for testing purpose from the whole set of instances of a sense of target word. Performance evaluation is measured in terms of precision and recall metrics. Precision is defined as the ratio of the correctly disambiguated instances and total number of test instances answered for a target word. Recall is defined as the ratio of the correctly disambiguated instances and total number of test instances to be answered for a target word. The translation, transliteration and details of the sense annotated Hindi corpus are given in Table A1 in Appendix.

Table 2. Sense annotated hindi corpus.

Number of Senses	Word
2	अशोक, कांड, कोटा, क्रिया, गल्ला, गुना, गुरु, ग्राम, घटना, चंदा, चारा, जीना, जेठ, डब्बा, डाक, ढाल, तान, ताव, तिल, तीर, तुलसी, दक्ष, दर, दाद, दाम, धन, धुन, बाल, माँग, लाल, विधि, शेर, सीमा, सोना, हल, हार
3	अंग, अंश, अचल, उत्तर, कदम, कमान, कुंभ, क्वार्टर, खान, चरण, तेल, थान, फल, मत, मात्रा, वचन, वर्ग, संक्रमण, संबंध
4	कलम, धारा, मूल
5	चाल, टीका

4.2. Experiments

In order to, evaluate corpus based lesk algorithm we conducted test runs by varying context window size of 5, 7, 10, 12 and 15. The average precision and recall over context window of 5, 7, 10, 12 and 15 is computed. The overall average precision and recall over 60 words is shown in Table 3. For evaluating WSD algorithm using word co-occurrence, test run is conducted on a fixed window size of 2. Precision and recall for 60 words is computed. We obtained an average precision of 68.73% and average recall of 64.41 % over all 60 words as shown in Table 4. For evaluating WSD algorithm using classification information model, test run is conducted on a fixed window size of 4. Precision and recall for 60 words is computed. We obtained an average precision of 76.34 % and average recall of 71.00 % over all 60 words as shown in Table 5. We have also listed the average precision and recall using direct overlap averaged over window size of 5, 10, 15, 20 and 25 of 60 words in Table 6. The direct overlap algorithm used in this paper is adapted from [19]. We remove stop words from sense definitions and test instances and then overlap is computed.

Table 3. Overall average precision and recall (corpus based lesk algorithm).

Weighting Scheme	Overall Average Precision	Overall Average Recall
TF	0.8429	0.7803
IDF	0.7842	0.7287
TF-IDF	0.8587	0.7954
CW	0.8510	0.7891

Table 4. Average precision and recall (WSD algorithm using word-co-occurrence).

Average Precision (Over 60 Words)	Average Recall (Over 60 Words)
0.6873	0.6441

Table 5. Average precision and recall (WSD algorithm using classification information model).

Average Precision (Over 60 Words)	Average Recall (Over 60 Words)
0.7634	0.7100

Table 6. Average precision and recall (direct overlap).

Average Precision (Over 60 Words)	Average Recall (Over 60 Words)
0.4787	0.4366

5. Results and Discussion

As shown in Table 3, overall average precision and recall using TF is 84.29% and 78.03%. Overall average precision and recall using IDF is 78.42% and 72.87%. Overall average precision using CW is 85.10% and 78.91%. The maximum overall precision and recall is obtained using TF-IDF which is 85.87% and 79.54%. The overall precision and recall using word co-occurrence probability is 68.73% and 64.41% respectively as shown in Table 4. The overall precision and recall using classification information model is 76.34% and 71.00% as shown in Table 5. The experimental results confirm that all the three algorithms perform significantly better than direct overlap method.

The TF of a word in a particular sense is high as compared to other sense. For example, the target word सोना (sona) has 2 senses: sleep and gold. The TF of चांदी (chandi/silver) (0.2692) is higher for सोना in gold sense as compared to सोना in sleep sense. This is because चांदी has more chances of occurring with सोना in gold sense rather than sleep sense. The TF of नींद (neend/sleep) (0.4615) is higher for सोना in sleep sense rather than gold sense because नींद is more likely to occur with सोना in sleep sense. This results in high accuracy of 84.29% when TF is used as a weighting function. IDF of a word is computed across all documents. IDF is low for words that occur more frequently in all documents. These words have less discriminatory power. The words that occur with a low frequency will have higher IDF value. This helps in identifying relevant document to a query in an information retrieval task. We expect this will help in sense discrimination. However, the observed accuracy using IDF alone is less as compared to using TF only. This is because the size of corpus in WSD task is not as large as in information retrieval task. Further, in a short window there is a little chance that for each sense we will be getting rare words that will have large IDF values as compared to other matching words to become dominant contributor to the score of a sense. Hence, we conducted a test run using TF-IDF as a weighting measure. The underlying assumption was that cases where a particular sense has strong but less frequent indicators will help in improving accuracy. Although we observed the maximum accuracy of 85.87% using this measure but the gain was not much as compared to using TF alone. The observed average precision using CWs is 85.10% which is quite close to the best performing case. The observed precision using second method is 68.73%. One of the reasons for poor performance as compared to first method may be the small window size used for collecting co-occurring words and phrases. We obtained a precision of 76.34% using the classification information model. The result is comparable with that obtained on English and Korean dataset. However this model suffers from data sparseness problem and knowledge acquisition bottleneck.

6. Conclusions

In this paper, we have evaluated and compared three Hindi WSD algorithms based on corpus statistics. These algorithms use a sense tagged training corpus to

gather the statistics. The corpus based lesk algorithm is a hybrid of supervised and knowledge based approach. We experimented with different weighting schemes for computing sense score and observed maximum precision of 85.87% using TF-IDF scheme. The WSD algorithm using co-occurrence probability yields a precision of 68.73%. The WSD algorithm using classification information model achieves a precision of 76.34%. All the three algorithms perform better than direct overlap between test instance and sense definitions. Based on our study we conclude the following:

1. The corpus-based statistics, if available, can be utilized in disambiguation.
2. TF collected over each sense helps significantly in improving the disambiguation performance.
3. If a sense annotated corpus is available, the conditional probability of co-occurring words and phrases can be pre-computed and utilized to advantage even in a limited setting.
4. The classification information model is language independent, easy to model and can exploit various types of clues for disambiguation. Further this model can be adapted for similarity-based approaches.

References

[1] Baldwin T., Kim S., Bond F., Fujita S., Martinez D., and Tanaka T., "A Reexamination of MRD-based Word Sense Disambiguation," *Journal of ACM Transactions on Asian Language Processing*, vol. 9, no. 1, pp. 1-21, 2010.

[2] Banerjee S. and Pederson T., "An Adapted Lesk Algorithm for Word Sense Disambiguation using WordNet," *in Proceedings of the 3rd International Conference on Computational Linguistics and Intelligent Text Processing*, Mexico City, Mexico, pp. 136-145, 2002.

[3] Banerjee S. and Pederson T., "Extended Gloss Overlaps as a Measure of Semantic Relatedness," available at: http://www.d.umn.edu/~tpederse/Pubs/ijcai03.pdf, last visited 2013.

[4] Hindi Corpus, available at: http://www.cfilt.iitb.ac.in/Downloads.html, last visited 2013.

[5] Hindi WordNet, available at: http://www.cfilt.iitb.ac.in/wordnet/webhwn/wn.php, last visited 2013.

[6] Kavitha A., "An Integrated Approach for Measuring Semantic Similarity between Words and Sentences using Web Search Engine," *the International Arab Journal of Information Technology*, vol. 12, no. 6, pp. 588-595, 2015.

[7] Khapra M., Bhattacharyya P., Chauhan S., Nair S., and Sharma A., "Domain Specific Iterative Word Sense Disambiguation in a Multilingual Setting," available at: http://core.ac.uk/download/pdf/23798934.pdf, last visited 2013.

[8] Khapra M., Joshi S., Chatterjee A., and Bhattacharyya P., "Together We Can: Bilingual Bootstrapping for WSD," *in Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, PA, USA, pp. 561-569, 2011.

[9] Khapra M., Shah S., Kedia P., and Bhattacharyya P., "Projecting Parameters for Multilingual Word Sense Disambiguation," *in Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Singapore, pp. 459-467, 2009.

[10] Kilgarriff A. and Rosenzweig J., "Framework and Results for English SENSEVAL," *Computers and the Humanities*, vol. 34, no. 1, pp. 15-48, 2000.

[11] Le C. and Shimazu A., "High WSD Accuracy using Naïve Bayesian Classifier with Rich Features," *in Proceedings of PACLIC 18*, Tokyo, Japan, pp. 105-113, 2004.

[12] Leacock C. and Chodorow M., *Combining Local Context and WordNet Sense Similarity for Word Sense Identification WordNet*, *An Electronic Lexical Database*, The MIT Press Cambridge 1998.

[13] Lee H., Baek D., and Rim H., "Word Sense Disambiguation based on the Information Theory," *in Proceedings of Research on Computational Linguistics Conference*, Taiwan, pp. 49-58, 1997.

[14] Lee H., Rim H., and Seo H., "Word Sense Disambiguation using the Classification Information Model," *Computers and the Humanities*, vol. 34, no. 1, pp. 141-146, 2000.

[15] Lee Y., Ng H., and Chia T., "Supervised Word Sense Disambiguation with Support Vector Machines and Multiple Knowledge Sources," *in Proceedings of the 3rd International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, Barcelona, Spain, pp.137-140, 2004.

[16] 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*, Ontario, Canada, pp. 24-26, 1986.

[17] Rezapour A., Fakhrahmad S., and Sadreddini M., "Applying Weighted KNN to Word Sense Disambiguation," *in Proceedings of the World Congress on Engineering*, London, UK, pp. 6-8, 2011.

[18] Sense Annotated Hindi Corpus, available at: http://www.tdil-dc.in/index.php?option=com_up-download&task=view-download-tool&view=download&toolid=1472, last visited 2013.

[19] Singh S. and Siddiqui T., "Evaluating Effect of Context Window Size, Stemming and Stop Word Removal on Hindi Word Sense Disambiguation," *in Proceedings of*

International Conference on Information Retrieval and Knowledge Management, Kuala Lumpur, Malaysia, pp. 1-5, 2012.

[20] Singh S., Singh V., and Siddiqui T., "Hindi Word Sense Disambiguation using Semantic Relatedness Measure," *in Proceedings of the 7th Multi-Disciplinary workshop on Artificial Intelligence*, Krabi, Thailand, pp. 247-256, 2013.

[21] Sinha M., Kumar M., Pande P., Kashyap L., and Bhattacharyya P., "Hindi Word Sense Disambiguation," available at: http://megha.garudaindia.in/iitb-nlp/hindiwn/papers/HindiWSD.pdf, last visited 2013.

[22] Suderman K., "Simple Word Sense Discrimination," *Computers and the Humanities*, vol. 34, no. 1, pp. 165-170, 2000.

[23] Turney P., "Word Sense Disambiguation by Web Mining for Word Co-occurrence Probabilities," available at: file:///C:/Users/ acit_pc/Downloads/5763802.pdf, last visited 2013.

[24] Vasilescu F., Langlasi P., and Lapalme G., "Evaluating Variants of the Lesk Approach for Disambiguating Words," available at: http://www.lrec-conf.org/proceedings/lrec2004/pdf/219.pdf, last visited 2012.

[25] Yang E., Zhang G., and Zhang Y., "The Research of Word Sense Disambiguation Method based on Co-occurrence frequency of Hownet," *in Proceedings of the 2nd Chinese Language Processing Workshop*, Hong Kong, China, pp. 60-72, 2000.

Satyendr Singh received BE degree in computer science and engineering from Ch. Charan Singh University, Meerut, India in 2000. He obtained ME in computer science and engineering from Panjab University, Chandigarh, India in 2008. Currently, he is pursuing PhD from University of Allahabad, Allahabad, India. His research interests include natural language processing, information extraction/retrieval, human computer interaction and machine learning.

Tanveer Siddiqui is currently Assistant Professor at University of Allahabad, Allahabad, India. She obtained M.Sc. and Ph.D degree in computer science from University of Allahabad. She has experience of teaching and research of more than 14 years in the area of computer science and information technology with special interest in natural language processing, human computer interaction and information extraction and retrieval.

Appendix

Table A1. Translation, transliteration and details of sense annotated hindi corpus.

Word	Sense Number : Translation Of Senses In English (Number Of Instances)
अंग (Ang)	Sense 1: Any Part or Organ of Human Body (88) Sense 2: Component (30) Sense 3: Part of a Community, Organization or Unit (105)
अंश (Ansh)	Sense 1: Numerator in Maths in Hindi (42) Sense 2: Component (36) Sense 3: Degree, Measurement of Angle (53)
अचल (Achal)	Sense 1: Immovable (12) Sense 2: Person's Name (34) Sense 3: Immovable Property (27)
अशोक (Ashok)	Sense 1: Name of a Tree in India (33) Sense 2: Name of an Indian King (21)
उत्तर (Uttar)	Sense 1: Answer (30) Sense 2: North Direction (79) Sense 3: A Person's Name (36)
कदम (Kadam)	Sense 1: Initiative (16) Sense 2: Foot (13) Sense 3: Step (11)
कमान (Kamaan)	Sense 1: Bow , Curved Piece of Resilient Wood with Taut Cord to Propel Arrows (28) Sense 2: Command (35) Sense 3: An Special Army (Eg, Navy) (33)
कलम (Kalam)	Sense 1: Pen, Quill (67) Sense 2: Cutting of a Tree (69) Sense 3: Style of Painting of a Particular Place (66) Sense 4: Place Near Ear and Cheeks , Where There are Hairs (26)
कांड (Kaand)	Sense 1: Part of Religious Literature (43) Sense 2: Negative Event or Happening (29)
कुंभ (Kumbh)	Sense 1: Waterpot Made of Mud (65) Sense 2: A Sun Sign (Aquarius) in Hindi (58) Sense 3: A Holy Event Happing Every 12 Years in India (64)
कोटा (Kotaa)	Sense 1: Reservation, Quota (70) Sense 2: Name of A District In Rajasthan in India (64)
क्रिया (Kriyaa)	Sense 1: Verb In Hindi Grammar (116) Sense 2: Activity, Action (71)
क्वार्टर (Quarter)	Sense 1: A Place Allotted To Live for Temporary Period (26) Sense 2: A Quantity of Wine (14) Sense 3: A Match, in Which After Winning, A Player or Team Reaches Semi Final (12)
खान (Khan)	Sense 1: Mine (60) Sense 2: Vast Storage of Subject Knowledge or Quality (13) Sense 3: Surname of A Muslim Community in India (65)
गल्ला (Galla)	Sense 1: Foodgrains (Wheat, Corn, Cereal) (41) Sense 2: Penny Bank, Piggy Bank (29)
गुना (Guna)	Sense 1: Times (22) Sense 2: Name of a District in Madhya Pradesh in India (21)
गुरु (Guru)	Sense 1:Teacher (89) Sense 2: Jupiter (Name of a Planet) (60)
ग्राम (Gram)	Sense 1: Village (169) Sense 2: A Unit Of Measurement, Gram (77)
घटना (Ghatnaa)	Sense 1: Event (65) Sense 2: Lowering Of Water Level, Subside (14)
चंदा (Chanda)	Sense 1: Moon (82) Sense 2: Financial Contribution, Subscription (75)
चरण (Charan)	Sense 1: Stage, Phase (72) Sense 2: Foot (49) Sense 3: Quarter Part of Anthology (78)
चारा (Chaaraa)	Sense 1: Domestic Animal's Food, Provender, Forage (100) Sense 2: Option (21)
चाल (Chaal)	Sense 1: Speed (13) Sense 2: Move to be Taken In Chess or Similar Games (97) Sense 3: A Place Where People Stay, Tenement House (11) Sense 4: Behavior (37) Sense 5: Strategy in Game, Trick (26)
जीना (Jeena)	Sense 1: To Live, Survive (39) Sense 2: Staircase (33)
जेठ (Jeth)	Sense 1: Name of a Month in Hindi (10) Sense 2: Husband's Elder Brother, Brother in Law (20)
टीका (Tika)	Sense 1: A Sign on Forehead Using Sandalwood (15) Sense 2: Vaccination (22) Sense 3: To Write About Something in Detail (24) Sense 4: A Ceremony to Confirm Marriage in India, Engagement Ceremony (10) Sense 5: A Jewelry Which is Worn by Indian Bride on Forehead (24)
डब्बा (Dabba)	Sense 1: Box , Made Up of Plastic, Wood or Metal, Bin (21) Sense 2: Coach of Train Which Carries Passengers (24)
डाक (Daak)	Sense 1: Bid, Bidding (60) Sense 2: Post, Postal System (59)
ढाल (Dhaal)	Sense 1: Sloping or Sliding Land (31) Sense 2: A Protective Covering Used for Saving Attack of Sword, Armour (28)
तान (Taan)	Sense 1: Process of Stretching (14) Sense 2: Music Tone (19)
ताव (Tav)	Sense 1: Torrid (18) Sense 2: Ream of Paper (8)
तिल (Til)	Sense 1: Sesame, a Plant From Which Oil Is Extracted From its Seeds (41) Sense 2: Mole (263)
तीर (Teer)	Sense 1: Arrow (103) Sense 2: Shore of River or Sea (39)
तुलसी (Tulsi)	Sense 1: Basil, a Plant Which is Considered Holy and Medicinal (193) Sense 2: A Saint Who was Follower of God Ram and Who Wrote Ramayana (81)
तेल (Tel)	Sense 1: Oil (128) Sense 2: Crude Oil Obtained From Mines (53) Sense 3: A Ceremony Performed In Indian Marriages (14)
थान (Thaan)	Sense 1: Roll Of Cloth, Bolt (21) Sense 2: A Place Where Domestic Animals Are Tied (9) Sense 3: Place Of Indian God Or Goddess (8)
दक्ष (Daksh)	Sense 1: A King in Indian Mythology Who was Father of Sati and Father in Law of Lord Shiva (64) Sense 2: Qualified, Efficient, Skilled (15)
दर (Dar)	Sense 1: Standard Cost, Rate (147) Sense 2: Door (67)
दाद (Daad)	Sense 1: To Praise Someone, Accolade (27) Sense 2: Skin Disease, Ringworm (51)
दाम (Daam)	Sense 1: Cost, Price (61) Sense 2: Type Of Strategy or Policy (20)
धन (Dhan)	Sense 1: Money , Wealth (126) Sense 2: Sign of Addition In Mathematics in Hindi, + (16)
धारा (Dhaaraa)	Sense 1: Law Charges for Crime In Indian Constitution, Section (44) Sense 2: River's Flow, Stream (67) Sense 3: Flow of Speech, Thought or Events (50) Sense 4: Electric Current (67)
धुन (Dhun)	Sense 1: Music Tune (84) Sense 2: Cult, Flakiness, Mania (10)
फल (Phal)	Sense 1: Fruit (90) Sense 2: Result (79) Sense 3: Front Sharp Part of Arrowor Spear (11)
बाल (Baal)	Sense 1: Hair (111) Sense 2: Child (47)
मत (Mat)	Sense 1: Religious Community (41) Sense 2: Opinion, Thought, Idea (31) Sense 3: Vote (92)
माँग (Maang)	Sense 1: Requirement, Need, (13) Sense 2: Parting of Hairs On Head Where Married Hindu Woman Put Vermilion As A Sign of Marriage (33)
मात्रा (Maatra)	Sense 1: Quantity , Amount , Volume (41) Sense 2: Some Time Period in Music (8) Sense3 :Vowel Sound in Hindi Speech (39)
मूल (Mool)	Sense 1: Root Of Plant (6) Sense 2: Basic Reason, Fundamental (49) Sense 3: Time for a Type of Star (97) Sense 4: Capital/Principal Money (40)
लाल (Laal)	Sense 1: Red Color (129) Sense 2: Son, Child (26)
वचन (Vachan)	Sense 1: Whatever One Speaks or Says, Saying (23) Sense 2: Promise, Commitment (27) Sense 3: Agent in Hindi Grammar to Denote Singular or Plural (23)
वर्ग (Varg)	Sense 1: Community, Category, Class (90) Sense 2: Square Object (15) Sense 3: Square of Number, Unit of Measurement of Area(E.G, Square Feet) (129)
विधि (Vidhi)	Sense 1: Way or Process of Doing Something (72) Sense 2: Law (69)
शेर (Sher)	Sense 1: Tiger, Lion (166) Sense 2: Type of Urdu Poetry (41)
संक्रमण (Sankraman)	Sense 1: Process of Sun's Transition From One Star-Sign to Another (28) Sense 2: Process of Disease Infection (60) Sense 3: Process of Transition From One Place or State to Another Place or State (22)
संबंध (Sambandh)	Sense 1: Relation (23) Sense 2: Agent In Hindi Grammar That Shows Relation Between Two Words (33) Sense 3: Marriage (8)
सीमा (Seema)	Sense 1: Limit, Threshold (28) Sense 2: Boundary, Border (23)
सोना (Sona)	Sense 1: Gold (65) Sense 2: Sleep (24)
हल (Hal)	Sense 1: Solution (26) Sense 2: Ploughing Instrument, Plough (76)
हार (Haar)	Sense 1: Defeat (33) Sense 2: Necklace, Garland (63)