

# Word sense disambiguation in Hindi applied to Hindi-English machine translation

**S Mall, U C Jaiswal\***

*Madan Mohan Malaviya University of Technology Gorakhpur, India*

*\*Corresponding author's e-mail: shachimall@gmail.com*

*Received 9 March 2017, www.cmnt.lv*

## Abstract

The Word Sense Disambiguation for Hindi Language is one of the biggest challenges faced by Natural Language Processing. In this paper we discuss issues in reducing ambiguity in Word Sense Disambiguation for Hindi Language. The concepts are induced in two modules Parsing and Word Sense Disambiguation for Hindi Language. Parsing is an extension of our previous work on shallow parser method that creates groups word which are essential for Machine Translation. Monolingual Hindi and English corpora are used. Following this we used machine learning technique such as supervised approach, unsupervised approach and domain specific sense with the help of Knowledge based methods. Knowledge based method uses Hindi and English WordNet tools. Supervised method is used to disambiguate the multiple tags in the context label with the correct tag. Unsupervised method is used to update the sentence with the correct sense and parts of speech tag. There are various websites which provide the facility of translation of Hindi language to English language such as Google Translator and Babefish Translator but these translators fail to resolve polysemy words in Hindi sentences the result is discussed in this paper. The accuracy result of part of speech tagging generated by our system is 92.09%. The accuracy results generated by our system for Chunk are window-3, window 2 and window1 are: 94.45%, 81.23%, and 81.11% respectively. We modify and develop Lesk algorithm which uses WordNet tools for Word Sense Disambiguation. We compare the system's performance with the website Google Translator. We also examine errors made by Google Translator for given input Hindi sentence. Our system generates correct translation with Word Sense Disambiguation for given input Hindi sentence as shown in the Figure12.

## Keywords:

Domain specific sense, Word Sense Disambiguation, Morphological analysis, Part of speech tagging and Parsing

## 1 Introduction

Word Sense Disambiguation in Hindi Language is difficult problem for finding the correct sense of a word in a context, when the word is polysemy. To identify the correct sense for machine translation is very difficult problem. This problem categorized in the field of Natural Language Processing [1]. There are many ongoing approaches is used to resolve Word Sense Disambiguation in machine translation for Hindi Language. We apply Rule based approach as well as machine learning techniques, which performs both unsupervised approach and supervised approach.

The objective is to resolve this ambiguity problem in Word Sense Disambiguation for Hindi language and produce correct translation in English Language for example **उस कबूतर के पर कतर दो** Here Hindi word **पर** has two synsets meaning. Synsets for adjective grammar **पर** means other and Synsets for noun grammar **पर** means wing. To find out Synsets meaning we have used English WordNet [5] and Hindi WordNet [6]. WordNet contain synset set of synonyms and ontological categories. Ontological categories consist of syntactic category like Noun, Pronoun, verb etc. We propose Lesk algorithm that uses WordNet tools for disambiguation has modified.

Development of good quality of machine translation for Hindi to English language using limited resource is challenging task. There are many website available for translation of Hindi language to English language such as Google translator [3] and Babefish Translator [4] but they

are fail to resolve polysemy Hindi word. The output result of Google translator is discussed in Table 1 and Table 2 discussed the result of Babefish Translation from Hindi to English language. Google Translation [3] and Babefish Translation [4] fail to resolve word sense disambiguation. Our developed system discussed in Table 3 resolves word sense disambiguation and produce correct translation from Hindi sentence to English sentence

TABLE 1 Google Translator

S. No.	Google Translator from Hindi language to English language
Input Sentence	उस कबूतर के पर कतर दो
Output Sentence	Two doves on the Qatar

TABLE 2 Babefish Translator

S. No.	Babefish Translator
Input Sentence	उस कबूतर के पर कतर दो
Output Sentence	The pigeon at Qatar two

TABLE 3 Our system generated translation from Hindi language to English language

S. No.	Our system generated translation from Hindi language to English language
Input Sentence	उस कबूतर के पर कतर दो
Output Sentence	The pigeon at Qatar two

The proposed work to develop Machine Translation System is divided into the following modules:

1. Morphological Analyzer

2. Parts of Speech Tagging
3. Chunk
4. Parsing
5. Word Sense Disambiguation
6. Hindi to English Translation

In our previous work [2] we had developed a system for first three modules. In this paper we developed for later three modules. In section 2, we describe different module definition in Section 3. Related work in Section 4. Proposed work in Section 5 Simulation Result and Analysis and Section 6 Conclusion and Future work

## 2 Preliminaries

Word sense disambiguation (WSD) is a primal problem for different Indian Languages Technology such as Machine Translation. We have developed Word sense disambiguation using parsing method for Hindi language and used this method for Hindi to English Translation. This paper proposed following methods:

### 2.1 DICTIONARY TO SENSE SEMANTIC CATEGORY

We use Hindi WordNet and English WordNet for disambiguation and translation. WordNet contain dictionary definition for each word and label with unique frequency ids. Disambiguation is performing on sentence by sentence basis. The frequency is manually tagged in domain table. Domain table is corpora contains with list of words and meaning of each word with their domain name. Domain name contain information of words belong to which category in given Hindi WordNet. In the Hindi WordNet word and their meanings are given it consist of following characteristics:

1. Synset
2. Gloss
3. Position in Ontology
4. Hyponymy and Hypernymy

Detail description of Hindi WordNet for Hindi word सोना  
word = ".सोना".decode('utf-8', 'ignore')

while True:

if word2Synset.has\_key(word):

synsets = word2Synset[word]

print "Word -->", ". सोना "

for pos in synsets.keys():

print "POS Category -->", pos

for synset in synsets[pos]:

print "\t\tSynset -->", synset

if synonyms.has\_key(synset):

print "\t\t\tSynonyms -->", synonyms[synset]

if synset2Gloss.has\_key(synset):

print "\t\t\tSynset Gloss", synset2Gloss[synset]

if synset2Onto.has\_key(synset):

print "\t\t\tOntological Categories", synset2Onto[synset]

if synset2Hypernyms.has\_key(synset):

print "\t\t\t\t\tHypernym\t\t\t\t\tSynsets",

synset2Hypernyms[synset]

if synset2Hyponyms.has\_key(synset):

print "\t\t\t\t\tHyponym\t\t\t\t\tSynsets",

synset2Hyponyms[synset]

word = raw\_input("Enter a word:").decode("utf-8",

"ignore")

### 2.2 MORPHOLOGICAL ANALYZER

Hindi language is morphology rich and free order in nature. Morphological information is used to constructed basic meaning units called morphemes. We identify the morphological information from tokenize words. The feature structure of Morphological Analyser is given below:  
<fsaf = 'root, lcat, gend, num, pers, case, vibh, suff'>

These eight cases are mandatory for the morph 'fs' is feature structure which contains 'af' is a composite attributes consisting of root of the word, Lexical category of the root, Gender of the word, Number corresponding to the word form, Person of the word, Case (Direct / Oblique), case name, Specificity Marker, Emphatic Marker, Dubitative Marker, Interjection Marker, Conjunction Marker, Honorific Marker, Gender of the agreeing noun, Number of the agreeing noun, Person of the agreeing noun. Form of suffix and prefix representing we take input from text file then apply suffix smoothing and prefix smoothing is done for example Identification of prefix token we extract feature of token. Token character are extracted up to character length seven (7) for example consider Hindi token अरे now each character of this token extracted अ1 अर2 अरे3 NULL4 NULL5 NULL6 NULL 7 feature extraction for suffix token characters up to length three (3) अरे रे1 अरे2 NULL3 Total word length is 3, this method is used to identify root word in the given context. The result of feature extraction is discussed in our previous paper [2] Morphological Analyzer example for Hindi token हिन्दी((NP<fsaf='हिन्दी,n,f,sg,3,d,0,0'head='hinxI'>हिन्दीNN P<fsaf='हिन्दी,n,f,sg,3,d,0,0' name='hinxI'>)).

### 2.3 PARSING

Parsing uncover the hidden structure of Hindi text input it can provides structural description that can identifies the break intonation and analyse a given sentence to determine its syntactical structure according to the part of speech tag and chunk. In natural language processing the syntactic analysis of Hindi language can vary from low level such as Part of speech tagging methodology has been discussed in our previous paper [2] Part of speech tagging is a process of labelling tags to each token with their related parts of speech such as nouns, verbs, adjectives, adverbs etc. to each word in the given sentence. We consider 19 Parts of speech class for Hindi language Table 4 shows the abbreviation classes of Parts of speech. To remove ambiguity in multiple tag for a single word we use Hidden Markov is also called Maximum Likelihood Tagger [5] for Parsing to identify the dependency between each predicate in a given input sentence. We use Viterbi approximation in equation (12) to choose the most probable tag sequence for given input Hindi sentence. To estimate we read off count from the training corpus and then computer the maximum likelihood. Firstly we calculate Transition matrix we have a set of words in a given sentence.

TABLE 4 Abbreviation classes of Parts of speech

S.No.	Symbol	Parts of speech
1	NN	Noun
2	NNS	Noun Plural
3	NST	Noun denoting spatial and temporal expressions
4	NNP	Proper Nouns
5	PRP	Pronoun
6	DEM	Demonstratives
7	VM	Verb Main
8	VAUX	Verb Auxiliary
9	JJ	Adjective
10	RB	Adverb
11	PSP	Postposition
12	RP	Particle
13	CC	Conjuncts
14	WQ	Question Words
15	QF	Quantifiers
16	QC	Cardinals
17	QO	Ordinals
18	SYM	Special Symbol
19	NEG	Negative Words

Figure 1 shows the flow chart of parts of speech tagging with following step

- User input Hindi sentence. The sentence is converts the input file into Shakti Standard Format (SSF) to Trigram to Shakti Standard Format (TnT).
- Build Transition Count Matrix and Build Emission count matrix. Build a hash of the tag sequence and its frequency calculated by equation 1
- N-grams smoothing technique is used as discussed in equation 8,9,10 and 11. The tag sequence of a given word sequence.
- Convert the output generated by part of speech tagger which is in TnT format to SSF format.

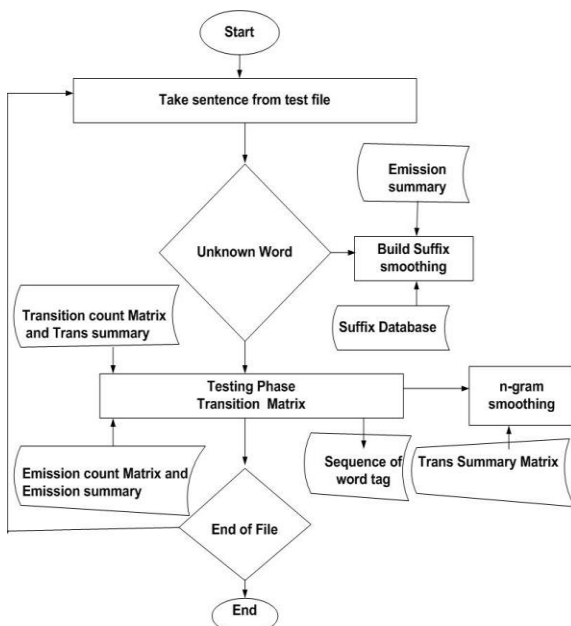


FIGURE 1 Flowchart and Description Process of Part of Speech Tagging

To remove ambiguity in multiple tag for a single word we use Hidden Markov is also called Maximum Likelihood Tagger [5] for Parsing to identify the dependency between each predicate in a given input sentence. We use Viterbi approximation in equation (12) to choose the most probable tag sequence for given input Hindi sentence. To estimate we read off count from the training corpus and then computer the maximum likelihood. Firstly we calculate Transition

matrix we have a set of words in a given sentence  $W_1-----W_T$  represents the sequence of the word. T is the probable tag sequence  $T = t_1, t_2, \dots, t_n$

$$\hat{T} = \arg \max_{T \in \tau} P(T / W) . \tag{1}$$

Equation (1) is used to choose the sequence of tags that maximizes  $\{P(T)P(W / T)\} / P(W)$

$$\hat{T} = \arg \max_{T \in \tau} \{P(T)P(W / T)\} / P(W) . \tag{2}$$

We make use of N-gram method for modelling the probability of word sequences. From the chain rule probability:

$$P(T)P(W/T) = \prod_{i=1}^n P(w_i/w_1t_1 \dots w_{i-1}t_{i-1})P(t_i/w_1t_1 \dots w_{i-1}t_{i-1}) . \tag{3}$$

Thus we are choosing the tag sequence that maximizes:

$$P(t_1)P(t_2/t_1) \prod_{i=3}^n P(t_i/t_{i-2}t_{i-1}) \prod_{i=1}^n P(w_i/t_i) , \tag{4}$$

$$\arg \max \left[ \prod_{i=1}^T P(t_i/(t_{i-1}, t_{i-2}))P(w_i/t_i) \right] P(t_{T+1}/t_T) . \tag{5}$$

We use Maximum likelihood estimation from relative frequency to estimate these probabilities:

$$P(t_i/t_{i-2}t_{i-1}) = c(t_i - 2t_{i-1}t_i) / c(t_i - 2t_{i-1}) , \tag{6}$$

$$P(w_i/t_i) = c(w_i, t_i) / c(t_i) . \tag{7}$$

Equation 2, 3, 4 and 5 calculate the probabilities of N-gram smoothing technique. This technique is used to resolve the issues of multiple tags for same word.

$$u \text{ nigrams} = \hat{P}(t_3) = \frac{f(t_3)}{N} , \tag{8}$$

$$B \text{ igrms} = \hat{P}(t_3/t_2) = \frac{f(t_2, t_3)}{f(t_2)} , \tag{9}$$

$$T \text{ rigrams} = \hat{P}(t_3/(t_1, t_2)) = \frac{f(t_1, t_2, t_3)}{f(t_1, t_2)} , \tag{10}$$

$$Lexical = \hat{P}(w_3/t_3) = \frac{f(w_3, t_3)}{f(t_3)} . \tag{11}$$

Let us consider an example Input Hindi Text: यह एशिया की सबसे बड़ी मस्जिदों में से एक है ।

The abbreviation of the Parts of speech tag is given in Table 1. Let user input Hindi sentence denoted by X and w is word tokenize from the sentence  $X = w_1, w_2, \dots, w_n$ .

Let each word in the given Hindi sentences is sequentially label with their corresponding Parts of speech tag  $T = T_1, T_2, \dots, T_n$ , where  $T_i \in T (i \leq i \leq n)$ .

Let S is a set of sequence tagging word with related tag

$$\begin{aligned} (u_1, u_2 \dots u_n, v_1 \dots v_n) \in S \quad P(u_1 \dots u_n, v_1 \dots v_n) \geq 0 \\ \sum P(u_1 \dots u_n, v_1 \dots v_n) = 1 \quad (u_1 \dots u_n, v_1 \dots v_n) \in S \end{aligned}$$

Hence  $(u_1 \dots u_n, v_1 \dots v_n)$  is a probability distribution over a pair of sequence set S. our approach Trigram Hidden Markov consists of a finite set W of possible words, and a finite set T of possible tags, together with the following parameters:

Sequence of pair  $(u_1 \dots u_n, v_1 \dots v_{n+1})$  such that  $n \geq 0, u_i \in W$  For  $i = 1 \dots n, u_i \in T$  For  $T_i = 1 \dots N$   
 $v_{n+1} = Stop$   
 Probability for any sequence

$$\begin{aligned} (u_1, u_2 \dots u_n, v_1 \dots v_{n+1}) \in S \quad \text{as } P(u_1 \dots u_n, v_1 \dots v_{n+1}) = \\ \prod_{i=1}^{n+1} q(v_i / v_{i-2}, v_{i-1}) \\ \prod_{i=1}^n e(u_i / v_i) \end{aligned}$$

where we assume u is a sentence and v is a tag

$$\begin{aligned} v_0 = v_{-1} \\ n = 11 \\ u_i \dots u_{11} \end{aligned}$$

यह एशिया की सबसे बड़ी मस्जिदों में से एक है ।  
 $q \langle JJ \rangle * \langle NP \rangle * \langle PSP \rangle * \langle QF \rangle * \langle JJ \rangle * \langle NNP \rangle * \langle PRP \rangle * \langle PSP \rangle * \langle QC \rangle * \langle VM \rangle * \langle SYM \rangle$  Here we use second order Markov Model  
 एशिया  $\langle NP \rangle$  की  $\langle PSP \rangle$  सबसे  $\langle QF \rangle$  बड़ी  $\langle JJ \rangle$  मस्जिदों  $\langle NNP \rangle$  में  $\langle PRP \rangle$  से  $\langle PSP \rangle$  एक  $\langle QC \rangle$  है  $\langle VM \rangle$  ।  $\langle SYM \rangle$  let a set of sentence  $u_1 \dots u_n$  and paired with sequence of tag  $v_1 \dots v_n$ .

Define  $l(a, b, c, d, e, f, g, h, i, j, k)$  to be the number of times to the sequence of 11 state is seen in training data  $\langle JJ, NP, PSP, QF, JJ, NNP, PRP, PSP, QC, VM, SYM \rangle$ . Similarly, define  $l(a; b)$  to be the number of times the tag bigram (a; b) is seen. Define  $l(s)$  to be the number of times that the state S is seen in the corpus can be interpreted as conditional probability where u is the sentence, यह एशिया की सबसे बड़ी मस्जिदों में से एक है । and v is a tag Maximum likelihood estimate are  $q(S / a, b) = l(a, b, \dots, n) / l(a, b)$ , where a, b, ...n are number of words in the given sentence.

$$e(u / S) = c(s \rightarrow u) / c(S),$$

where  $c(s \rightarrow u) =$  number of time state S is seen paired with observation in the corpus  $c(v \rightarrow यह)$  would be the number of times the word यह is seen paired with the tag v

$$q(JJ, NP, PSP, QF, JJ, NNP, PRP, PSP, QC, VM, SYM) = \frac{c(JJ, NP, PSP, QF, JJ, NNP, PRP, PSP, QC, VM, SYM)}{c(JJ, NP, PSP, QF, JJ, NNP, PRP, PSP, QC, VM, SYM)}$$

$$e(यह / v) = c(v \rightarrow 2Tag) / c(v)$$

To estimate we read off count from the training corpus and then computer the maximum likelihood. Firstly we calculate Transition matrix we have a set of words in a given sentence.

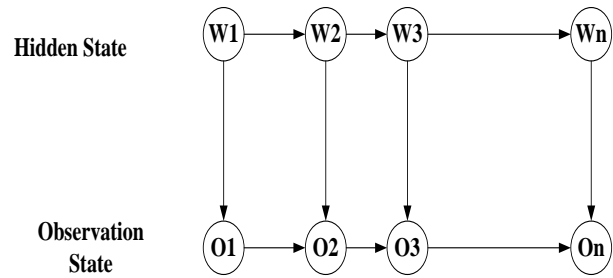


FIGURE 2 Transition matrix method

एशिया  $\langle NP \rangle$  की  $\langle PSP \rangle$  सबसे  $\langle QF \rangle$  बड़ी  $\langle JJ \rangle$  मस्जिदों  $\langle NNP \rangle$  में  $\langle PRP \rangle$  से  $\langle PSP \rangle$  एक  $\langle QC \rangle$  है  $\langle VM \rangle$  ।  $\langle SYM \rangle$

The process starts from one word to another word. Each move is called a step. If the chain is currently in state X i then it moves to state X i at the next step with a probability denoted by P i,j, and this probability does not depend upon which states the chain was in before the current state this method is known as Transition matrix Figure 2 elaborate the method to calculate Transition matrix  $Q_i = [W_i = i]$ . Suppose we have N-state Hidden Markov Model parameterized by (E, Q, R) where emission probability represents E, Q is an initial probability and transition probability matrix represent R. Let rows of R identical and given by vector r, the joint probability of the hidden states and observations over a sequence of length O can be  $O = O1, O2, O3 \dots On$  Calculated as:

$$\begin{aligned} Z(U, V) / E, Q, R ) \\ = Z u_1 / Q \prod_{o=2}^O Z(u_i / R(u_{i-1}, :)) Z(v_o / u_o, E ) \\ = Z(u_1 Q \prod_{o=2}^O Z(u_o, r) Z(v_o / u_o, E) \quad (12) \end{aligned}$$

Maximum likelihood can calculate the sequence  $\lambda = (W_{i,j}, O_{i,j}, Q_i)$   
 Output POS Tag: यह  $\langle JJ \rangle$  एशिया  $\langle NP \rangle$  की  $\langle PSP \rangle$  सबसे  $\langle QF \rangle$  बड़ी  $\langle JJ \rangle$  मस्जिदों  $\langle NNP \rangle$  में  $\langle PRP \rangle$  से  $\langle PSP \rangle$  एक  $\langle QC \rangle$  है  $\langle VM \rangle$  ।  $\langle SYM \rangle$

Through the above calculation we find tag for other words in a given sentence and input for the process of Chunk. Figure 11 shows the snapshot of Parsing with Parts of speech tagging for given Hindi sentence. Chunking [3] is an important process to identifying and segmenting the text into syntactically correlated chunk tag such as is NP chunk label the word in the sentence start with different Phrases, we label the word with boundary marker B represents - Beginning phrase and I represent as Inside phrase for example we input Hindi sentence: दफ्तर के सभी लोग अपनेअपने घरों को जाने की जल्दी में थे।

दफ्तर NN B-NP के PSP I-NP सभी QF B-NP लोग NN I-NP अपने PRP B-NP SYM I-NP अपने RDP I-NP घरों NN B-NP

The sentence is individually tokenize by the delimiter “?” in sentence start with  $\langle Sentence\ id=? \rangle$  chunk start with assigning chunk number “((chunk phrase=Hindi word, the features of the word and chunk Table 5 shows the abbreviation of chunk symbols. Chunk is an arbitrating step towards parsing. In Figure 3 Head computation is used for functional specification to

compute the phrase with heads of different phrases of groups such as noun, verb groups etc. Chunk head provides the sufficient information for further processing of the sentence. Figure 7 shows the output result of Hindi token label with related parts of speech tagging and chunk.

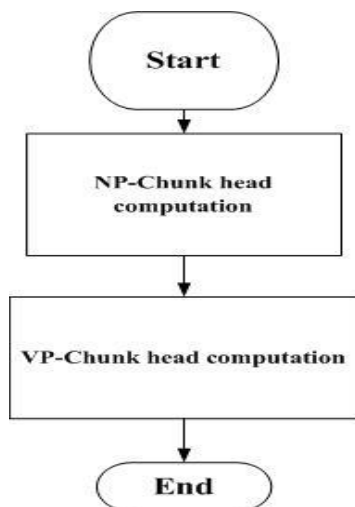


FIGURE 3 Chunk computation

## 2.4 WORD SENSE DISAMBIGUATION

Hindi words have more than one meaning in the context. for example Hindi word 'सोना' hold English meaning gold and सोना hold English meaning sleep we have use English WordNet synset and Hindi WordNet synset. Synset contains list of synonyms. Hindi WordNet is an ontological category. Ontological categories are coarse grained distinctions of word senses. Our methods for word sense disambiguation are following

- Unsupervised approach: Unsupervised method used to find the correct word in polysemy word and update the sentence.
- Supervised approach: Supervised method used to sense from a sense tagged corpora
- Domain Information for Sense Disambiguation: We use WordNet for domain information.
- Modified Lesk algorithm: Modified Lesk algorithm used for the target word's synset which has maximum overlap of its gloss, its hypernymy gloss and its hyponymy gloss with the words in the context of target word is chosen as the sense of the word

Ambiguity can be resolved with syntactic information. Word Sense ambiguities [3] disambiguate the senses of word with the meaning of multi-sense words using Distributed Domain approach by analyzing the context in which sense the multi-sense words and produce correct output. In Hindi language contains multi-sense words in its corpus. This is based on supervised and unsupervised approach [4]. Supervised approach [5] is used to identify the correct meanings in multi-sense words in Hindi languages. If the database does not contain sufficient information then it cannot sense the ambiguous word. Unsupervised approaches use dictionary for learning and shows the correct result and update the database. This can be done by using WordNet [6,7,8,9] as lexical database resource to identify the correct meanings in multi-sense words in Hindi languages. Domain specific Modified Lesk's algorithm [10]

is based on dictionary definitions which are also called glosses. This technique is a resource as a key factor to word senses in corpora. Where sense disambiguation is performed using the overlap between the contexts in which a word appears in the discourse.

## 2.5 HINDI TO ENGLISH TRANSLATION

We find a mapping to an English word for each sense in Hindi and predict the translation of a polysemy Hindi word in an English context. A mapping is available between Syset identifier in English Wordnet and Hindi sense dictionary.

## 3 Related work

A number of Indian researchers have carried out their work related to machine translation for Indian languages. Little work has been carried out in Hindi language. Sense disambiguation is done by using dictionary method [11] without using prior annotated data. New linear time algorithm for lexical chaining [12] is used for word sense disambiguation. Sense dependency and selective dependency [13] using this combination word sense disambiguation problem was resolved. Various technique have been proposed by researchers to resolve the issues like Unsupervised Weighted Graph [14] the result was 54.9 % in sensaval-3 and 60.2 % in sensaval-2 dataset. Dependency parsing approach is a dynamic programming based algorithm [15]. Knowledge based [16] is a multilingual joint approach for Word Sense Disambiguation. Graph based Word Sense Disambiguation is used to improve the performance of both monolingual Degree and PLength, and compete with the state of the art on all disambiguation tasks. [17] Word Sense Disambiguation use dominant senses of words in specified domains. The accuracy values of approximately 65%. They presented the methodology for Word Sense Disambiguation based on domain 23information [18]. The drawback of the algorithm is that it can disambiguate a word provided it has only one sense per domain. The work on English French Cross-lingual Word Sense Disambiguation [19] French to English translation the target English word depending on the context by identifying the nearest neighbours of the test sentence from the training data using a pairwise similarity measure. The performance of the system was less than the baseline by around 3% it outperformed the baseline system for 12 out of the 20 nouns [20].

## 4 Proposed work

This paper is extension of my previous paper [2] in which morphological analysers and parts of speech tagging is completed Using these modules we develop we developed algorithm for Parsing and Word Sense Disambiguation which are as below:

### 4.1 PARSING

#### Algorithm 1 Parsing

1. for  $i \leftarrow 0$  to length words
2. do
3. for each word is Chunk with Noun phrase then
4. Select parent head word "B"
5. Select part of speech H

6. Select voice of H
7. Select position of H (left, right)
8. Else if word is a verb then
9. Select nearest word N to the left word such that word is the parent head word of "I"
10. Select nearest word r to the right of word such that word is the parent head word of r
11. Select part of speech of l
12. Select part of speech r
13. Select the part of speech word
14. Select voice of word
15. Else if word is adjective then
16. Select parent head word head
17. Select part of speech of head
18. end

Parsing is used to estimate the number of useful probability concerning and its syntactical structure of the sentence the method is explained in section 2 parsing. In the parsing algorithm we develop some identification rule are as follows:

- In case of NN most of the time ambiguity is in case marking (direct, oblique). We can decide the case on the basis of following PSP.
1. Rule 1: If NN is just followed by PSP, then we will take only the feature structures having oblique case. Else we will take the direct case.
  - In case of JJ the case (d/o), should agree with the noun it is modifying.
  2. Rule 2: If JJ has multiple morph analysis then we will look for noun it is modifying and we will take the morph analysis of JJ having case marked same as that of modified noun and eliminate the rest.
  - In case of PSP the pruning module is giving multiple morph analysis for 'ke' and 'ki'.
  3. Rule 3: We will look for the noun to which our PSP is related and will keep the morph analysis having gender and case agreeing with the gender and case of the noun to which our PSP is related and eliminate the rest.

Most of the time the noun is related with PSP is found in the next chunk to chunk containing PSP. Then most probably head of the chunk is NP.

#### 4.2 WORD SENSE DISAMBIGUATION

##### Algorithm 2 Modified Lesk Algorithm

Input: Text with only meaningful words

Output: Actual sense of ambiguous words

1. Loop Start for all dictionary definition of the ambiguous word
2. Ambiguous word is selected
3. Each word is selected from preliminary input texts.
4. Gloss of ambiguous word is obtained from typical WordNet.
5. Intersection is performed between the meaningful words from the input text and the glosses of the ambiguous word.
6. Loop End
7. If the counter value is mismatched with all other values, then associated sense is considered as the disambiguated sense.
8. Else, Bag-of-Words fails to disambiguate the sense.
9. If occurrence of an unmatched word in anticipated database having a particular sense crosses the

threshold value, then the word is moved to the related bag of words database.  
10. Stop.

##### 4.1.1 Definitions

Let window of context is,  $2t+ 1$  with the grammar R. Were list of the word in WordNet is define  $T_i, 1 \leq i \leq R$ . compare the list of in the WordNet is less than  $2t + 1$ , if all the list of words in WordNet belong to the context. Were  $T_i$  is list of words contain more than two meaning in the gloss, list of words are assign with unique synset having a unique sense tag. Were  $T_i$  is the lists of sense tag are represented by  $|T_i|$ . We evaluate sense tag for each pair of words in the context of the window. Were  $R = \sum |T_i|$  represents combinations of words this is referred as candidate combination.

##### 4.1.2 Process

When user input the sentence then each word are tokenize and label with grammatical tag. If the tag is multiple then it depends on tokens of the input sentence. Words which is polysemy is labeled with multiple tag their related parts of speech here we use N gram technique. We divide the context in three windows window 1, window 2 and window 3. Window 1 applies unigram method take only right word next to the polysemy word and finds the candidate combination as given in equation 2. Window 2 applies bigram method take only left word next to the polysemy word and finds the candidate combination as given in equation 3. Window 3 applies unigram method take right and left both words next to the polysemy word and finds the candidate combination as given in equation 3. we use overlap technique discussed in the flowchart of Figure 3. Overlap technique find overlap between pair of word with polysemy word in each window. We map each combination with gloss if the combination scores high then we consider that word given in the gloss with their related tag and update the list with the help of unsupervised approach.

For example Input Hindi sentence: उस कबूतर के पर क़तर दो.

Two glosses can have more than one overlap where each overlap covers as many words as possible. Each gloss compares with pairs of words divide the sentence in three size window. Each window has pair of relation we find individual score by comparing the combination score with particular candidate as shown in the Table 4. After comparing highest score window will winner. Winner word will be chosen as correct sense for given polysemy word पर Hindi sentence. The target word, are specified by WordNet. The window of the sentence would be पर क़तर दो and उस ,के token are separated as shown in Table 5, Table 6, and Table 8. Table 8 shows all word from Table 5, Table 5 and Table 7 possible pair Finally, two senses of the keyword "पर" have their counter readings (refer Table 5) as follows:

$$P \text{ counter, } PC = E' + F' + U' + S'$$

$$Y \text{ counter, } YC = E'' + F'' + U'' + S$$

TABLE 5 Sense for Token पर

Keyword	sense
पर	P
	Q

TABLE 6 Sense for token कतर

Context Word	sense
कतर	E
	F

TABLE 7 Sense for token दो

Context Word	Likely sense
दो	U
	S

TABLE 8 Candidate pairs of the word in the given sentences

Candidate pairs	General word in the sentence
P and A	E
P and B	F'
Q and A	E''
Q and B	F''
P and U	U'
P and S	S'
Q and U	U''
Q and S	S''
P and E	E'
P and F	F'
Q and E	E''

The Hindi word पर meaning is nothing more in a sentence and other meaning of पर is wing of a bird. The list of words is stored and their meaning is stored in the list of words can find the correct sense of a word having different meaning due to different contexts. The list of words sense the disambiguate word which is considered as keyword. The general words are separated from the sentence only keyword and context word is compared with each word of each "sense" list of words searching for the maximum frequency of words in common. The above algorithm is based on the learning set. In initial stage, if word is not present in the learning set, then it will not participate for disambiguation. Though, its probable meaning would be stored in the database. When the number of occurrences of the particular word with a particular sense crosses specific threshold value, the word is inserted in the learning set to take part in disambiguation procedure. Therefore, the efficiency of the disambiguation process is increased by this auto increment property of the learning set. Output Hindi sentence translated in English sentence: Slice/cut/chop of the wings

TABLE 9 Precision, Recall and F-score for parts of speech tagging

Abréviation of parts of speech	Precision %	Recall%	F-Score	Accuracy
CC	95.75	98.235294	97.909091	94.33333333
DEM	63.157895	92.307692	75	60
INJ	80	36.363636	50	33.33333333
JJ	69.230769	66.176471	67.669173	51.13636364
NEG	100	100	100	100
NN	76.352705	96.455696	85.234899	74.26900585
NNP	100	27.272727	42.857143	27.27272727
NST	100	100	100	100
PRP	91.891892	80	85.534591	74.72527473
PSP	97.154472	95.6	96.370968	92.99610895
QC	85.714286	100	92.307692	85.71428571
QF	75	81.818182	78.26087	64.28571429
RB	100	42.857143	60	42.85714286
RP	84.090909	88.095238	86.046512	75.51020408
SYM	100	100	100	100
VAUX	91.715976	85.635359	88.571429	79.48717949
VM	80.269058	85.238095	82.678984	70.47244094
WQ	89.473684	89.473684	89.473684	80.95238095
OVERALL SYSTEM	89.655647	92.862734	91.717502	94.97284249

of that pigeon.

#### 4.2 HINDI TO ENGLISH TRANSLATION

##### Algorithm 3 Translation of Hindi sentence to English sentence

Input: Hindi word tagged with corresponding English word  
 Output: Translation of Hindi word in English word in the context.

1. Each word in the context is in the root form.
2. Extract each word from the sentence.
3. Generate unique ids to each Hindi word and store in the database
4. Map each Hindi word with the English WordNet dictionary.
5. Return the label of the selected English translation of the target Hindi word in the context.
6. Stop.

For translation of Hindi sentence into English sentence we use the concept of mapping between English WordNet and Hindi sense dictionary. The given Hindi word is searched in the Hindi sense dictionary for synset frequency ids of all the synset. We use these sense ids to query the English WordNet to label all the synset. The set of all English words are mapped with the unique ids of the Hindi words. The English words contained in the English synset ids for translation of Hindi words to English words.

#### 5 Simulation result and analysis

The simulations have been carried out using Python language to obtain the accuracy results of parts of speech tagging and Chunk. Chunking is a method used for parsing the Hindi sentences. The evaluation result of Precision, Recall and F-score for parts of speech tagging is discussed in Table 9 and Chunk evaluation results is discussed in Table 10, 11 and 12. The output result of system generated parts of speech tag and Chunk are compared with Gold Standard parts of speech tag and Chunk [2]. Gold standard contains correct output of the parts of speech tag and Chunk for the given words. The total Hindi token was 1657 tokens with 990 phrases. Label each token with related parts of speech.

5.1 PARTS OF SPEECH TAGGER

This is simulation result of parts of speech tagger. The data set value is taken for 1657 tokens within 990 phrases and found correct parts of speech tag for 1024 tokens within 919 phrases. Figure 4 shows the graph plot for Parts of speech tagger with the help of evaluation results of Precision, Recall, F score and accuracy data is as below:

- Precision 'P' = Correct POS / (Correct POS + False POS)
- Recall 'R' = Correct POS / (Correct POS + False POS)
- F- Score or measure it evaluate the test accuracy by computing the value of both the precision P and the recall R:
- F-score 'FB1' =  $2 * RP / (R+P)$
- Accuracy 'A' =  $R * P / (R+P+F)$

Results for each part of speech were calculated in confusion matrix. Confusion matrix is shown in Figure 5 shows the correct match by system generated with Gold standard. The comparison result of system generated with Gold standard part of speech is given below:

TP: 1408 Counter({'NN': 381, 'PSP': 239, 'VM': 179, 'SYM': 166, 'VAUX': 155, 'PRP': 68, 'JJ': 45, 'RP': 37, 'CC': 30, 'DEM': 24, 'NNP': 18, 'WQ': 17, 'NST': 15, 'NEG': 12, 'QF': 9, 'QC': 6, 'INJ': 4, 'RB': 3, 'QO': 0, 'INJC': 0, 'RDP': 0, 'QCC': 0, 'NNPC': 0, 'RBC': 0, 'VMC': 0, 'JJC': 0, 'ECH': 0, 'PRPC': 0, 'NNC': 0})

FN:248 Counter({'NN': 118, 'VM': 44, 'JJ': 20, 'VAUX': 14, 'DEM': 14, 'PSP': 7, 'RP': 7, 'PRP': 6, 'NNC': 6, 'QF': 3, 'NNPC': 3, 'CC': 2, 'WQ': 2, 'INJ': 1, 'QC': 1, 'NNP': 0, 'QO': 0, 'NEG': 0, 'RB': 0, 'INJC': 0, 'RDP': 0, 'NST': 0, 'QCC': 0,

'RBC': 0, 'VMC': 0, 'JJC': 0, 'ECH': 0, 'PRPC': 0, 'SYM': 0})  
 FP: 248 Counter({'NNP': 48, 'VM': 31, 'NNC': 29, 'VAUX': 26, 'JJ': 23, 'PRP': 17, 'NN': 14, 'PSP': 11, 'INJ': 7, 'RDP': 6, 'RP': 5, 'RB': 4, 'CC': 4, 'NNPC': 4, 'INJC': 3, 'QO': 2, 'QF': 2, 'WQ': 2, 'DEM': 2, 'VMC': 'NEG': 0, 'NST': 0, 'QC': 0, 'SYM': 0}) Accuracy: 92.09%; precision: 84.76%; recall: 89.29%; F-score: 86.97%.

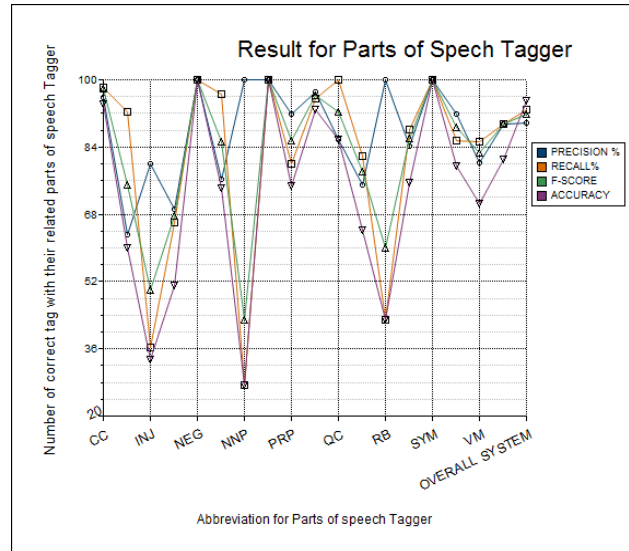


FIGURE 4 Precision, Recall, F-score & Accuracy for parts of speech tagger

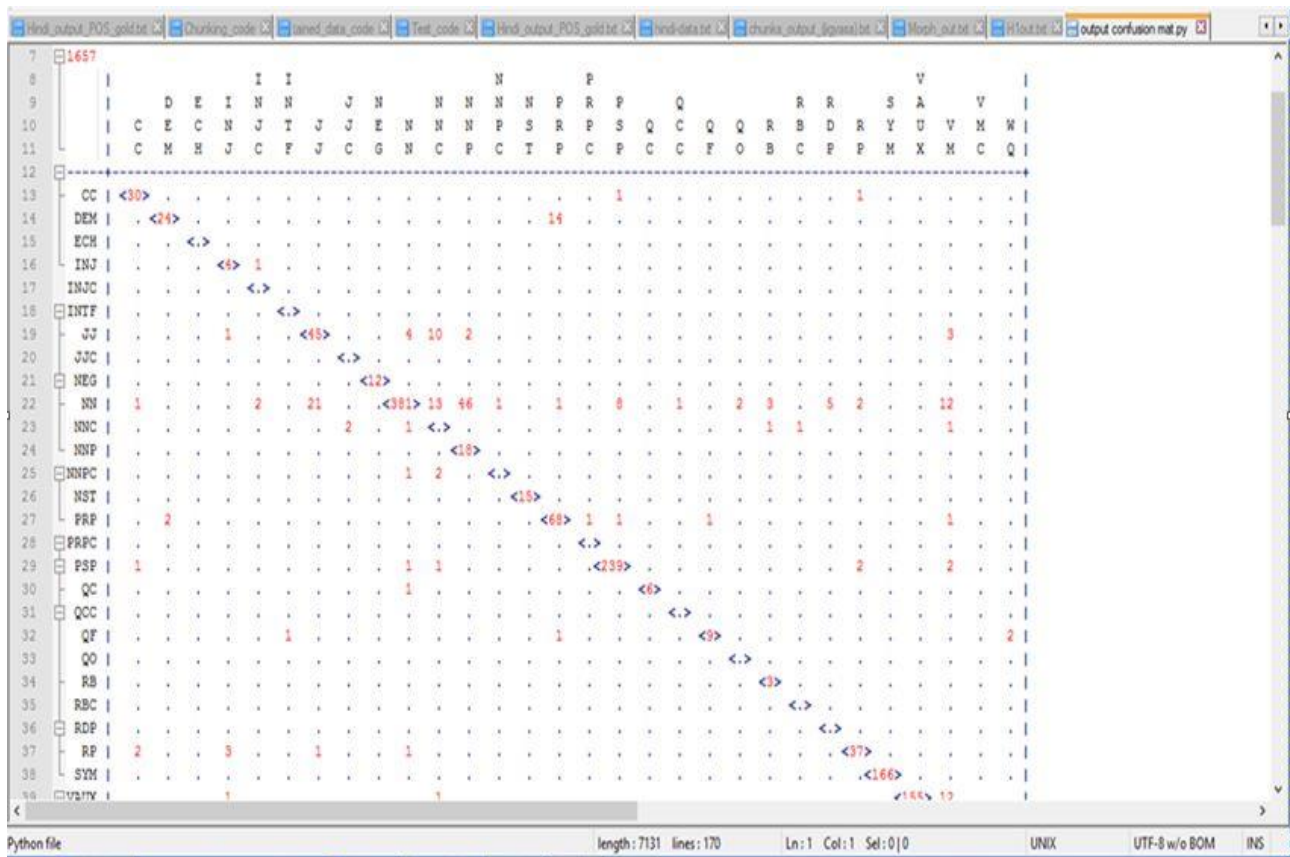


FIGURE 5 Precision, Recall, F-score & Accuracy for parts of speech tagger evaluated through confusion matrix



5.2 CHUNK

We created confusion matrix for Chunk as same way as we created for parts of speech tag. Figure 6 shows the graph and Figure 7 shows the output snapshot of chunk from given input Hindi sentence. The evaluation results for chunk is calculated for Precision, Recall, F-score and Accuracy for chunk are divided into three windows such as window 1, window 2, and window 3 in which we calculate Precision, Recall, F-score and Accuracy for each window. The data is as below:

TABLE 10 Window 1

	PRECISION	RECALL	FB1	ACCURAC
BLK	95.97	98.62	97.28	
CCP	100	100	100	
FRAGP	0	0	0	
JJP	62.96	58.62	60.71	
NEGP	50	100	66.67	
NP	91.51	95.65	93.53	
RBP	81.25	86.67	83.87	
VGf	87.5	97.47	92.22	
VGNF	78.57	61.11	68.75	
VGNN	46.67	43.75	45.16	
OVERALL SY	89.75	92.83	91.26	94.45

TABLE 11 Window 2

	PRECISION	RECALL	FB1	ACCURAC
BLK	95.97	98.62	97.28	
CCP	100	100	100	
FRAGP	0	0	0	
JJP	65.52	65.52	65.52	
NEGP	100	100	100	
NP	92.63	95.65	94.12	
RBP	76.47	86.67	81.25	
VGf	90.06	97.47	93.62	
VGNF	83.33	69.44	75.76	
VGNN	66.67	62.5	64.52	
OVERALL SY	91.24	93.64	92.42	95.17

TABLE 12 WINDOW 3

	PRECISION	RECALL	FB1	ACCURAC
BLK	95.33	98.62	96.95	
CCP	100	100	100	
FRAGP	0	0	0	
JJP	65.52	65.52	65.52	
NEGP	100	100	100	
NP	91.96	95.29	93.59	
RBP	81.25	86.67	83.87	
VGf	89.53	97.47	93.33	
VGNF	80	66.67	72.73	
VGNN	64.29	56.25	60	
OVERALL SY	90.67	93.23	91.93	94.75

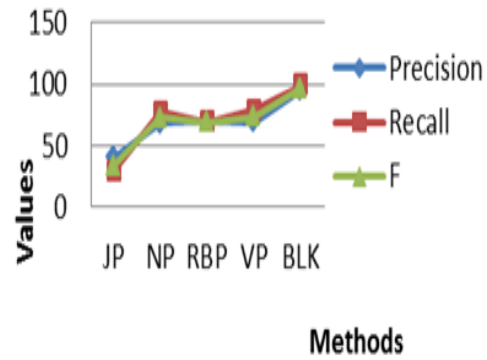


FIGURE 6 Accuracy result for Chunk for Window 1, 2 and 3

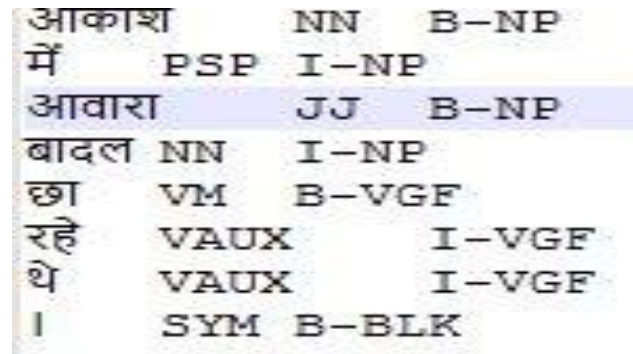


FIGURE 7 Chunk for Hindi sentence

5.3 WORD SENSE DISAMBIGUATION

For word sense disambiguation and translation we compare our system generated output of Hindi sentence to English sentence translation with translating website Google translator [21] as shown in the Figure 9 we input Hindi sentence तुम्हारे लिए मेरा दर खुला रहेगा। Here word दर in the given Hindi sentence is polysemy word which has two meaning Rate and Door. We compare our output result with translating website Google translator we input same Hindi sentence as shown in Figure 9. Here Hindi word दर is translated as Rate in the English sentence but correct translation is Door for the given sentence. Figure 8 shows that Google translator is failed to translate correctly but our system generates correct translation for the given Hindi word दर is translated as Door in the English sentence as shown in the Figure 12. We have input 100 Hindi sentence with polysemy words The result accuracy WSD systems generated output is compared with gold standards created by human annotators we have develop simulation method in which three candidate files in which first file collect the output result of system generated WSD as shown in Figure 12, second gold file which contains correct translation with WSD for given Hindi input sentence and the third file contain the output of Google Translator. Figure 8 shows the comparison graph with the system generated file and Google file with gold file to calculate the accuracy of WSD.

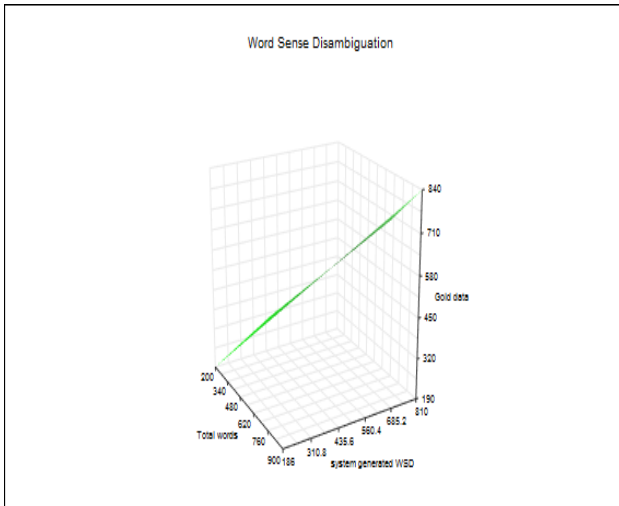


FIGURE 8 WSD systems generated output is compared with gold standards

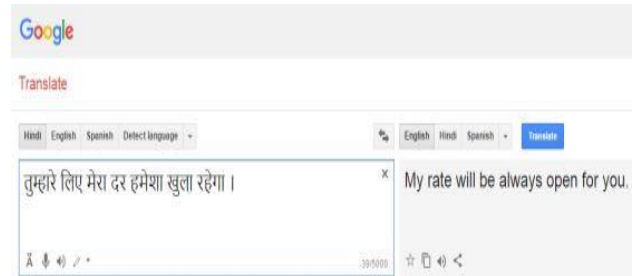


FIGURE.9 Google output of Hindi to English Translation

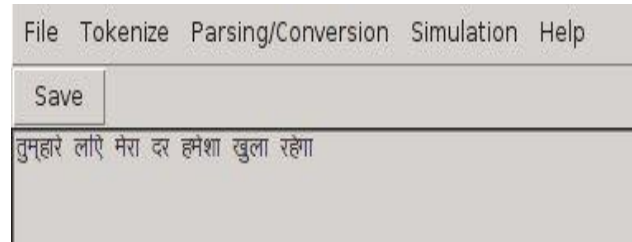


FIGURE 10 Input Hindi sentence with polysemy word दर

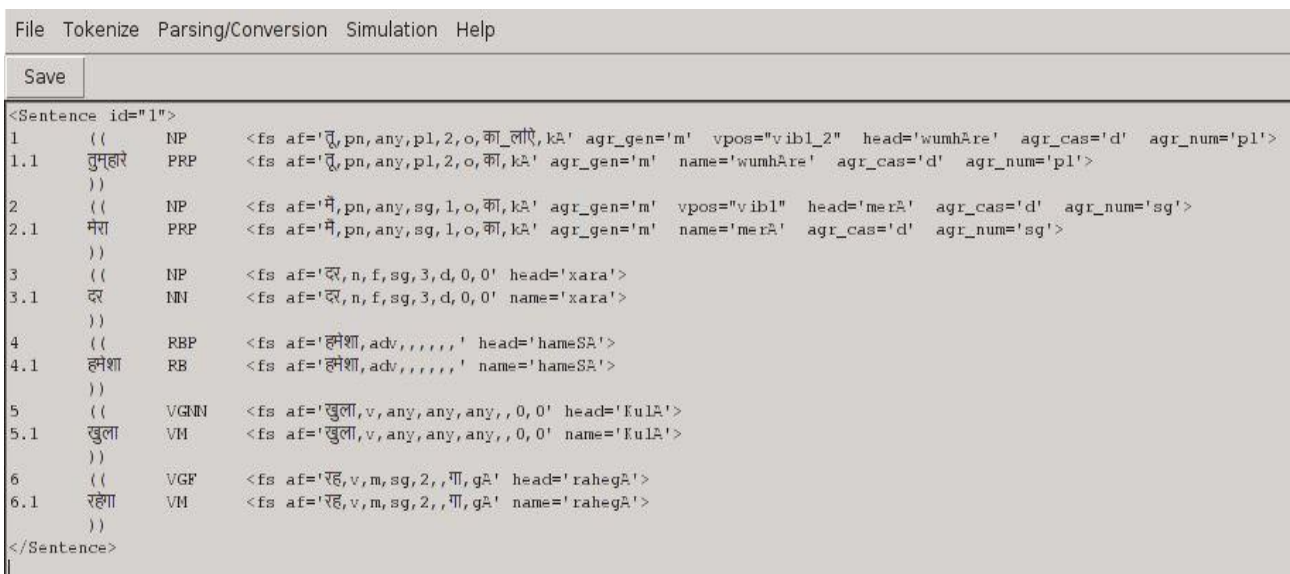


FIGURE 11 snapshot of Parsing for given Hindi sentence

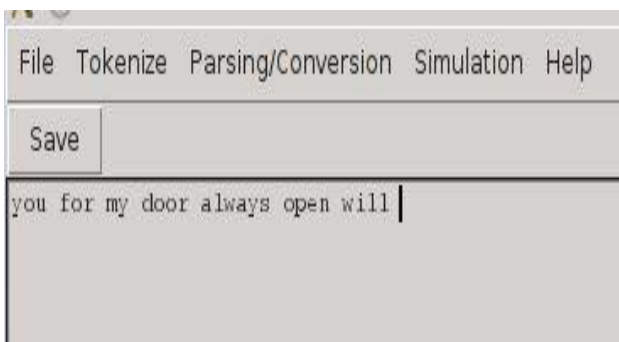


FIGURE 12 system generated output snapshot of Hindi to English Transaltion

**6 Conclusion and future work**

This work is carried out to evaluate how parsing is used for machine translation for Hindi language to English language. Figure 11 shows the output snapshot of Hindi sentence is parsed. No previous attempts have been reported in



literature, which analyze the Hindi Language translation into English translation for Indian language. This work confirms that the Parts of speech tagging algorithm obtain 92.09% accuracy result. The accuracy results for chunk are evaluated for three windows. Window-3, window 2 and window1 are: 94.45%, 81.23%, and 81.11% respectively. We enhance the Modified Lesk algorithms in which overlap is finding between three pieces of words in a given context to find the correct word sense by counting word overlaps between glosses of the words in the context. All the glosses of the key word are compared with the glosses of other words. The sense for which the maximum number of overlaps occur, represents the desired sense of the of the polysemy word. We use Hindi and English WordNet which is used in lexical knowledge. The Modified Lesk algorithm improves word sense disambiguation and the system generated result as shown in figure 12 is compared with Google translator website as shown in the Figure 9. This work shows that Google Translator cannot handled word sense disambiguation but our system can resolve word sense

disambiguation. We compare our system generated output with available translating website such as Google Translator. The result accuracy WSD systems is compared with gold standards created by human annotators we have develop simulation method in which three candidate files in which first file collect the output result of system generated WSD, second gold file which contains correct translation with WSD for given Hindi input sentence and the third file contain the output of Google Translator. We compare the

system generated file and Google file with gold file to calculate the accuracy of WSD. Our system can resolves word sense disambiguation and generate translation for each Hindi polysemy word in the given sentence without word alignment. This work can be further extracted by resolving the issues of language in which subject object and verb appear. Hindi language is subject verb object and English language is subject object verb.

## References

- [1] Chan Yee Seng, HweeTou Ng, David Chiang 2007 Word sense disambiguation improves statistical machine translation *In Annual Meeting-Association for Computational Linguistics* 45(1) 33
- [2] Mall Shachi, Umesh Chandra Jaiswal 2016 Evaluation for POS tagger, chunk and resolving issues in word sense disambiguate in machine translation for Hindi to English languages *In Computing for Sustainable Global Development (INDIACom), 2016 3rd International Conference on*, pp. 14-18.IEEE
- [3] Radu F, Cucerzan S, Schafer C, Yarowsky D 2002 Combining classifiers for word sense disambiguation *Natural Language Engineering* 8(04) 327-41
- [4] Eneko A, Edmonds P, eds. 2007 Word sense disambiguation: Algorithms and applications 33 Springer Science & Business Media
- [5] Mallapragada Pavan Kumar, Rong Jin, Anil K. Jain, Yi Liu 2009 Semiboost: Boosting for semi-supervised learning *IEEE transactions on pattern analysis and machine intelligence* 31(11): 2000-2014
- [6] Redkar Hanumant Harichandra, Sudha Baban Bhingardive, Diptesh Kanojia, Pushpak Bhattacharyya 2015 World WordNet Database Structure: An Efficient Schema for Storing Information of WordNets of the World *In AAAI* 4290-1
- [7] F Radu, S Cucerzan, C Schafer, D Yarowsky 2002 Combining classifiers for word sense disambiguation *Natural Language Engineering* 8(04) 327-41
- [8] Lesk M 1986 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 ACM* 24-6
- [9] Fellbaum C 1998 A semantic network of English verbs *WordNet: An electronic lexical database* 3 153-78
- [10] Resnik P 1995 Using information content to evaluate semantic similarity in a taxonomy *arXiv preprint cmp-lg/9511007*
- [11] Gaume B, Nabil Hathout, P Muller 2004 Word sense disambiguation using a dictionary for sense similarity measure *In Proceedings of the 20th international conference on Computational Linguistics* 1194 Association for Computational Linguistics
- [12] Galley M, McKeown K 2003 Improving word sense disambiguation in lexical chaining *In IJCAI* 3 1486-8
- [13] Chaplot Devendra Singh, Pushpak Bhattacharyya, Ashwin Paranjape 2015 Unsupervised Word Sense Disambiguation Using Markov Random Field and Dependency Parser *In AAAI* 2217-23
- [14] Hessami Ehsan, Faribourz Mahmoudi, Amir Hossien Jadidinejad 2011 Unsupervised Weighted Graph for Word Sense Disambiguation *In 2011 World Congress on Information and Communication Technologies*
- [15] Li Zhenghua, Min Zhang, Wanxiang Che, Ting Liu, Wenliang Chen, Haizhou Li 2011 Joint models for Chinese POS tagging and dependency parsing *In Proceedings of the Conference on Empirical Methods in Natural Language Processing* 1180-91 Association for Computational Linguistics
- [16] Navigli R, Ponzetto S P 2012 Joining forces pays off: Multilingual joint word sense disambiguation *In Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning* 1399-410 Association for Computational Linguistics
- [17] Khapra Mitesh, Pushpak Bhattacharyya, Shashank Chauhan, Soumya Nair, Aditya Sharma 2008 Domain specific iterative word sense disambiguation in a multilingual setting *In Proceedings of International Conference on NLP (ICON 2008), Pune, India*
- [18] Kolte Sopan Govind, Sunil G Bhirud 2008 Word sense disambiguation using wordnet domains *In 2008 First International Conference on Emerging Trends in Engineering and Technology* 1187-91 IEEE
- [19] Mahapatra Lipta, Meera Mohan, Mitesh M Khapra, Pushpak Bhattacharyya 2010 OWNS: Cross-lingual word sense disambiguation using weighted overlap counts and wordnet based similarity measures *In Proceedings of the 5th International Workshop on Semantic Evaluation* 138-41 Association for Computational Linguistics
- [20] Sawhney Radhike, Arvinder Kaur 2014 A modified technique for Word Sense Disambiguation using Lesk algorithm in Hindi language *In Advances in Computing, Communications and Informatics (ICACCI, 2014 International Conference on 2745-9 IEEE*
- [21] <https://translate.google.co.in/?hl=en>

AUTHORS	
	<p><b>Shachi Mall, 23-01-1986, India</b></p> <p><b>Current position, grades:</b> Teaching Cum Research Scholar fellowship            PhD[(CSE) [Thesis Submitted] M.M.M.U.T., Gorakhpur, M Tech(CSE) M.M.M.E.C., Gorakhpur, B. Tech(CSE)I.T.M., Gorakhpur  <b>Scientific interest:</b> Natural Language Processing  <b>Publications:</b> 12  <b>Experience:</b> 06</p>
	<p><b>Umesh Chandra Jaiswal, 01-06-1967, India</b></p> <p><b>Current position, grades:</b> Associate Professor  <b>University studies:</b> PhD(CSE),M Tech(CSE) IITD, BE(CE) M.M.M.E.C., Gorakhpur  <b>Scientific interest:</b> Natural Language Processing, Design and Analysis of Algorithms, Operating Systems, Computer Networks  <b>Publications:</b> 20  <b>Experience:</b> 25</p>