

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/260734205>

Measuring context-meaning for open class words in Hindi language

Conference Paper · August 2013

DOI: 10.1109/IC3.2013.6612174

CITATIONS

16

READS

460

3 authors:



Amita Jain

Ambedkar Institute of Advanced Communication Technologies and Research

64 PUBLICATIONS 520 CITATIONS

[SEE PROFILE](#)



Sudesh Yadav

Govt of Haryana

11 PUBLICATIONS 54 CITATIONS

[SEE PROFILE](#)



Devendra Kumar Tayal

IGDTUW

57 PUBLICATIONS 567 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Neutrosophic Set [View project](#)



Metaphor Processing [View project](#)

Measuring Context-Meaning for Open Class Words in Hindi Language

Amita Jain

Assistant professor, Dept. of
Computer Science & Engineering
Ambedkar Institute of Advanced
Communication Technologies and
Research,
Delhi-110031, India
amita_jain_17@yahoo.com

Sudesh Yadav

Dept. of Computer Science &
Engineering
Ambedkar Institute of Advanced
Communication Technologies and
Research,
Delhi-110031, India
yadavsudesh01@gmail.com

Devendra Tayal

Head, Dept. of Computer Science &
Engineering
Indira Gandhi Delhi Technical
University for Women,
Delhi – 110006, India
dev_tayal2001@yahoo.com

Abstract –Word sense disambiguation (WSD), the task of identifying the intended sense of words has been a growing research area in the field of natural language processing. In this paper, the authors focused on word sense disambiguation for Hindi language using graph connectivity measures and Hindi WordNet[1]. To construct the graph for the sentence each sense of the ambiguous word is taken as a source node and all the paths which connects the sense to other words present in the sentence are added. The importance of nodes in the constructed graph are identified using node neighbor based measures (various centrality) and graph clustering based measures (denseness, graph randomness, edge density). The proposed method disambiguates all open class words (noun, verb, adjective, adverb) and disambiguates all the words present in the sentence simultaneously.

Index Terms—Word sense disambiguation, Hindi Language, Hindi WordNet, Graph Connectivity Measures, Centrality.

I. INTRODUCTION

Word sense disambiguation (WSD), the task of identifying the correct sense of words in given context is the growing research area in natural language processing. Sense disambiguation was considered as an essential fundamental task for many computational applications such as machine translation, information retrieval, question answering, text summarization, intelligent data retrieval and speech recognition [2].

Most of the people in India use Hindi as their primary language and the language spoken by these people is ambiguous i.e. many words can be interpreted in multiple ways depending upon the context. To disambiguate words, machines need to process unstructured textual information and transform them into data structures (tokenization) for analyzing intended meaning [3]. The automatic identification of meanings of words by machines is called word sense disambiguation.

The difficulty in analyzing the meanings in intended context by machines arise due to lack of intelligent resources, presence of different domain and different sense inventories Turing test [4]. In this paper, for accomplishing WSD, authors refer to the graph based WSD. The constructed graph is used to evaluate the importance of vertices in the graph by measuring

connectivity measures based on node neighbor and graph clustering. A vertex having most “important” sense is considered to be the correct sense in context of intended meanings. Then results are simulated by experimental comparison of node neighbor based and graph clustering based methods. The method presented here disambiguates all open class words (noun, verb, adverb and adjective) and disambiguates all the words present in the sentence simultaneously.

This paper comprised of three parts: a general framework for Hindi Graph Based WSD (building graph, evaluating centrality measures III, IV), comparison results (V) and finally influence of the graph based method on Hindi Word Sense Disambiguation(VI).

II. RELATED WORK

WSD was initially formulated by Weaver [3] as a distinct task of computational linguistic in the late 1940s. In computational linguistic, WSD is the open problem [5]. In the early days of 21st century, researchers, [6] proposed a novel classifier combination approaches for word sense disambiguation task using Naive Bayes, Cosine, Decision List, Transformation-based Learning and MMVC classifiers. Researcher, [7] developed a state- of- art WSD system which integrates first time a WSD system performance with MT system performance. Scholars, [8] gave comparative evaluations using several measures of word semantic similarity metrics and graph centrality algorithm which lead to a performance competing with the state of art in unsupervised word sense disambiguation. Recently Robert Navigli[3], survey on word sense disambiguation approaches i.e. supervised, unsupervised and knowledge based approaches and proposed a graph connectivity measures based approach for unsupervised word sense disambiguation[9].

For WSD in English language a lot of work has been done but for Hindi the little research has been contributed. Initially, Bhattacharyya et al.[10] ,at IIT Bombay proposed the Hindi words sense disambiguation approach by comparing the linguistic context of the words in a sentence with the context constructed by Hindi WordNet using similarity based

approach, which works for nouns only. Tanveer J.Siddiqui [11], proposed an unsupervised approach to WSD by learning a decision list using untagged instances. Stemming has been applied and stop word has been removed then list is used for annotating an ambiguous word with its correct sense it also works for noun only. Researchers [12], proposed “Word Sense Disambiguation for Hindi Language”, resolves the ambiguity by making the comparisons between the different senses (nouns only) of the word in the sentence with the words present in synsets from the Hindi WordNet and the information related to these words in the form of parts-of speech. Authors [13], developed an approach for disambiguating ambiguous Hindi postposition by taking the problem with the case study of Hindi Punjabi machine translation. Thus disambiguation will be done on the basis of machine translation point of view. N-gram algorithm is used for Hindi postposition and extracting words from the corpus. Researchers [14], presents comparative evaluations on the basis of “word semantic similarity based graph based approach for unsupervised WSD”. Scholars [15], proposed two unsupervised approaches namely Flat Semantic Category Labeler (FSCL)[15] and Hierarchical Semantic Category Labeler (HSCR) [15]for unsupervised semantic category labeling for Hindi WSD by using ontological categories defined in Hindi WordNet as inventory. The approaches treat semantic categories as flat files and exploit the hierarchy among the semantic categories in a top down manner respectively. Researchers [16], gave performance Comparison of Word Sense Disambiguation (WSD) Algorithm on Hindi Language and the algorithm given by them based on Highest Sense Count and works well with Google. The objective of that paper is the comparative analysis of the WSD algorithm results on three Hindi language search engines- Google, Raftaar and Guruji and method is tested on a sample of 100 queries to check the performance of the WSD algorithm on various search engines. Satyendr Singh and Tanveer J.Siddiqui [17], evaluates the effects of stemming, stop word removal and size of window on a manually created sense tagged corpus consisting of Hindi words (nouns). Researchers [18] , proposed a genetic algorithm based approach for Hindi WSD. It also works for nouns only. In sum authors can say that approaches explained here for Hindi WSD works for nouns only.

III. GRAPH BASED WSD

In order to derive the effect of graph centrality connectivity measures on WSD, authors develop an algorithm which relies on graph structures for interpreting word senses. The construction of co-occurrence graph is based on grammatical relation described in Hindi WordNet. Throughout this paper, we assume that different word senses for each word are generated by reference lexicon and words are sense tagged.

A. Lexical database (Hindi WordNet)

Hindi WordNet is the online lexicon reference system which is designed by centre for Indian language technology solutions, IIT Bombay, inspired by usability of Hindi language in all over the world. It contains noun, verb, adjectives and

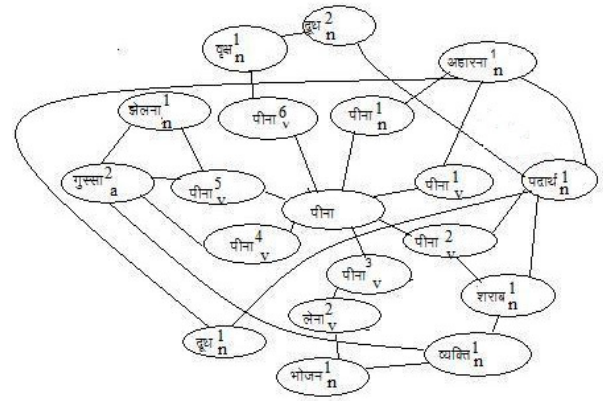


Fig. 1. An excerpt of Hindi WordNet graph centred around word पीना

adverb sense of a word, represented by synonyms (synsets). Here we discuss Hindi lexical database with an example- पीना which have six sense of verb, “पीना”, “लेना”, “खाना” etc.. These are represented as:-

- a). 1. पीना_v¹, पदार्थ_n¹
2. पीना_v², शराब_n¹
3. पीना_v³, खाना_n¹¹

Each word in lexical database is associated with part of speech tagged, with a subscript: n stands for noun, v stands for verb, a for adjective and r for adverb and superscript denoting sense number (i.e. पीना_v³ has third sense of the verb पीना_v). Sense numbering of a word is given by frequency of occurrence of that word in SemCor corpus. Each synset in WordNet is associated with a gloss: textual definition which explains its meaning.

Hindi WordNet lexical database have many relations: nominalization, hypernymy, pertainy, holonymy and hyponymy relations and so on.

- b). 1. पीना_v¹ $\xrightarrow{\text{Gloss}}$ दूध_n¹
2. पीना_v² $\xrightarrow{\text{Gloss}}$ शराब_n¹

Here gloss gives the textual meaning of a word.

Here in above figure 1. , we show an excerpt of Hindi WordNet graph centered on the word पीना_v using Hindi WordNet as resource.

Here adjacent vertices in the graph came from relations described in Hindi Wordnet, which are connected after performing depth first search strategy for all senses of word. Note that graph constructed here is undirected and not encoded with different kinds of relations.

B. Measuring context meaning

The proposed algorithm for measuring context meaning for Hindi word sense disambiguation works on the basis of target sentences. For which first we build a graph $G = (V, E)$, in which V corresponds to the sense of a word and E (edges)

connects pair of words which are syntactically related. To construct the graph for the sentence each sense of the ambiguous word is taken as a source node and all the paths which connects the sense to other words present in the sentence are added. In this paper, authors focused on words only (i.e. nouns, verbs, adjectives and adverbs) as explained in section I. Given graph G we consider the most appropriate sense S_i in context of sentence, listed in Hindi WordNet.

Algorithm for Measuring Context Meaning:-

Let W represents the given sentence

$W = (w_1, w_2, \dots, w_n)$ where w_i represents the i^{th} word in the sentence W where $1 \leq i \leq n$. To construct the graph from Hindi WordNet the following steps are followed:

- Find all senses associated with each word w_i ;
- Let $V_\pi = \bigcup_{i=1}^n \text{senses}(w_i)$ set V_i contains all possible senses of the words in the sentence W.
- For the graph G set $V := V_\pi$ and $E = \emptyset$;
- For each node $v \in V_i$, Perform depth search taking v as source node up to $\text{length} \leq L$, if we encounter a node v' such that $v' \in V_\pi$ and $v' \neq v$. Add all nodes and edges that encountered in the path from v to v' to the graph G. $V_\pi = V_\pi \cup \{\text{all the nodes that encountered in the path}\}$
- $E = E \cup \{\text{all the edges that encountered in the path}\}$;

With the help of above algorithm, we explore all syntactically related nodes up to $\text{edges} \leq 4$. But still some edges remain unexplored, for this we “backtracks” the same procedure to explore the remaining edges.

Let us explain graph construction process with an example:- वह दूध पीती है। Here the target words for our graph are दूध and पीना (i.e. $W_i = \text{दूध}_n, \text{पीना}_v$). First we take पीना as our target word, which have six verb senses ($\text{पीना}_v^1, \text{पीना}_v^2, \text{पीना}_v^3, \text{पीना}_v^4, \text{पीना}_v^5, \text{पीना}_v^6$).

In Hindi Wordnet पीना_v^1 is associated to अहारना_v^1 with causative relation, पीना_v^1 and पीना_v^2 to सेवन_n^1 with hypernymy relation, पीना_v^1 is also associated to पदार्थ_n^1 with gloss relation, so connect these words.

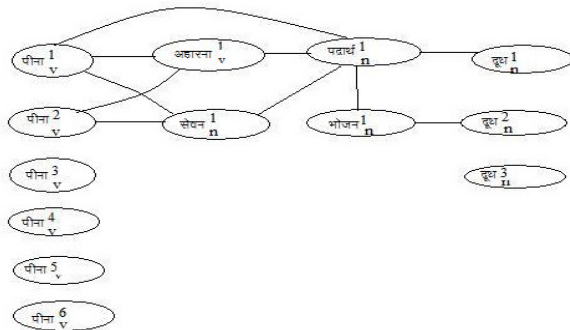


Fig. 2. Graph for the sentence वह दूध पीती है । ($W_i = \text{दूध}_n, \text{पीना}_v$).

DFS does not found any related senses to words $\text{पीना}_v^3, \text{पीना}_v^4$ and पीना_v^6 in context to text in the Hindi WordNet so we left these words. In this way using depth first search method we construct a graph with the help of Hindi WordNet, which is shown above.

With the help of above graph we next evaluate, which one of the senses of पीना and दूध are important in context with given sentence in graph. For this, we accomplish many graph connectivity measures (i.e. based on node neighbors' and graph clustering) which are discussed in next section.

IV. GRAPH CONNECTIVITY MEASURES

In this section, authors discuss different graph connectivity measures for finding which sense of target word is most important in context to given text. Although our graph connectivity measures can be applied for both directed and directed graph, in context to WSD. But here we are considering it for undirected graph.

A. Connectivity measures based on node neighbors

Local measures (i.e. based on node neighbors) are used for determining which vertex of graph is most important in context to given sentence.

$$n : V[1,0]. \quad (1)$$

A value in node neighbor based measure of a vertex near to one indicates that vertex node is important whereas value near to zero indicate vertex is peripheral. We evaluate here different types of node neighbor based measures (degree, edge betweenness centrality, eigenvector centrality and closeness centrality).

a) Degree centrality

Centrality is a function of the importance of a node in a graph [19]. Simplest type of centrality is the degree centrality which measures the value of a node by counting number of neighbors terminating from a given node.

$$\text{deg}(v) = |\{v, u\} \in E : u \in V| \quad (2)$$

$$C_{\text{deg}}(v) = \text{deg}(v) / (|V| - 1) \quad (3)$$

A node having highest value of degree centrality is called important, otherwise it is called peripheral. Conversely, a disconnected node always has degree centrality equals to zero.

According to fig. 2, degree centrality of पीना_v^1 is 0.25, which is highest among all senses of पीना. Thus in the given context first sense of पीना (i.e. पीना_v^1) is important because it has highest degree centrality among all senses of पीना_v and other nodes are peripheral nodes.

b) Edge Betweenness centrality

Edge between-ness centrality measures the importance of a node in a graph, by calculating the number of shortest paths from one node to another that passes through v.

$$E.\text{Betweenness}(v) = \sum_{r,t \in V : r \neq t \neq v} w_{r,t}(v) / w_r \quad (4)$$

$$C_{E.\text{Betweenness}} = E.\text{Betweenness}(v) / \{(|V| - 1) (|V| - 2) \} \quad (5)$$

Authors evaluate edge between-ness centrality for all nodes of a graph. A vertex which is having highest edge between-ness centrality[19] means that it is having large number of paths compared to the total sets of graph. A disconnected vertex always has edge between-ness centrality equals to zero as no path can pass through this node.

In Table I. पीना_v¹ and पीना_v² have same edge between-ness centrality. So there is a tie for पीना. In this case authors choose randomly. The edge betweenness centrality scores for all vertices are shown below in table I.

a) Centrality based on HITS and Page rank

Centrality based on HITS and Page rank is also called eigenvector centrality[19]. These centrality measures are the version of degree centrality. Degree centrality measures number of neighboring nodes a vertex has, whereas eigenvector centrality proves that not all connections (neighbors) are equal.

A weighted Page rank was first introduced by Mihalcea [20], which considers equal weights on edges when computing the scores associated with a vertex in a graph. Authors applied it for undirected graph here.

$$PR(v) = (1-d)/|V| + d \sum_{u \in V} (PR(u)/\text{Outdegree}(u)) \quad (6)$$

Where d is called damping factor. The above given formulae is for connected vertices and, if a vertex is disconnected then its page rank is given by first term of (6) (i.e. (1-d)/|V|). The value of Page rank of a node in a graph can be higher than one.

Hyperlinked – induced topic search (HITS) is another graph centrality algorithm that was designed for ranking Web pages according to their degree of “authority” and “hubs” (Kleinberg 1999) S. Brin [21][22]. Where authorities correspond to good reliable nodes that have numerous incoming links and hubs correspond to numerous outgoing links. Weighted version of HITS was introduced by Mihalcea [20][19], which consider edge weights for calculating hubs and authorities scores.

TABLE I. VALUES OF CONNECTIVITY MEASURES BASED ON NODE NEIGHBORS FOR VERTICES IN FIG. 2

Vertex	Degree	Edge Betweenness	PR	HITS	KPP
पीना _v ¹	0.25	0.08	0.75	0.35	0.40
पीना _v ²	0.16	0.08	0.58	0.23	0.33
पीना _v ³	0.00	0.00	0.01	0.00	0.08
पीना _v ⁴	0.00	0.00	0.01	0.00	0.08
पीना _v ⁵	0.00	0.00	0.01	0.00	0.08
पीना _v ⁶	0.00	0.00	0.01	0.00	0.08
अहारना _v ¹	0.25	0.03	0.89	0.27	0.40
सेवन _n ¹	0.25	0.04	0.89	0.27	0.40
पदार्थ _n ¹	0.42	0.05	1.03	0.48	0.50
भोजन _n ¹	0.16	0.04	1.18	0.14	0.36
दूध _n ¹	0.05	0.00	0.17	0.12	0.05
दूध _n ²	0.05	0.00	0.17	0.06	0.04
दूध _n ³	0.00	0.00	0.01	0.00	0.08

$$HITS_A^W(V_i) = \sum_{v_j \in \text{In}(V_i)} w_{j,i} HITS_H^W(V_j) \quad (7)$$

$$HITS_H^W(V_i) = \sum_{v_j \in \text{Out}(V_i)} w_{i,j} HITS_W^A(V_j) \quad (8)$$

Difference between HITS and Page rank is that one is evaluated on a sub-graph of relevant nodes whereas other evaluates the entire graph structure respectively.

Here in table I. page rank and HITS values for the above given sentence are given. Page rank gives a tie for दूध_n (which remains ambiguous), whereas HITS unequally selects पीना_v¹ and दूध_n¹ as best senses for sentence वह दूध पीती है।. In the context of given sentence HITS gives fine-grained sense ranking as compare to Page rank.

b) Closeness centrality

With closeness centrality, a vertex is considered important by measuring how close to all other nodes in a given graph. It is determined by reciprocal of shortest distance between a vertex to all others vertices [19]. Here closeness centrality is given by Key Player Problem.

$$C.C(v) = \sum_{u \in V: u \neq v} (1/d_{u,v}) / (|V|-1) \quad (9)$$

KPP for disconnected graph is given by 1/|V| (i.e. C.C (पीना_v³, पीना_v⁴, पीना_v⁵, पीना_v⁶, दूध_n³) = 0.08).

Here in above table best scores for sense पीना_v and दूध_n for C.C are shown by italic bold values.

B. Connectivity measures based on graph clustering

Graph clustering centrality measures are concerned with the whole graph structure irrespective of individual nodes. In this paper, authors discuss three well known graph clustering centrality measures (i.e. denseness, graph randomness and edge density) by dividing constructed graph in four clusters of different sense.

a) Denseness

Denseness measure measures the importance of clusters by cross referencing. A cluster having more denseness can be reached easily from other vertices [19].

$$De(G) = (\max - \sum_{u \in V} \sum_{v \in V} d(u,v)) / (\text{Max} - \text{Min}). \quad (10)$$

Where Max=K.|V| (|V|-1), is the maximum distance for disconnected graph and Min=|V| (|V|-1) the minimum sum value for fully connected graph. Here value of K is one when graph is fully connected otherwise zero (for disconnected graph). A graph cluster having maximum value for denseness is considered to be best structured cluster. Table II. gives the denseness values for different clusters (Fig. 4) of graph Fig 3.

a) Graph randomness(Entropy)

Graph randomness measures the amount of information stored in a random variable [19]. For evaluating randomness, first divide the main graph in clusters (e.g. distribution shown in Fig. 4). Then quality of each cluster is given by their randomness.

$$R(G) = - \sum_{v \in V} p(v) \log(p(v)) \quad (11)$$

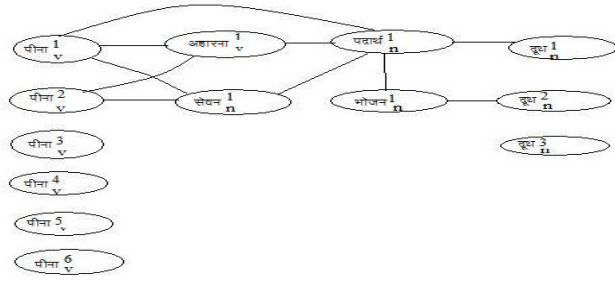


Fig. 3. Graph for the sentence वह दूध पीती है । ($W_i = \text{दूध}_n, \text{पीना}_v$).

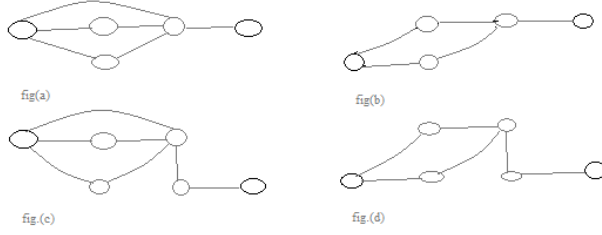


Fig. 4. Graphs for global measures corresponding to meanings to given context.

TABLE II. VALUES OF CONNECTIVITY MEASURES BASED ON GRAPH CLUSTERING APPLIES TO FOUR GRAPHS REPRESENTING THE MEANING OF GIVEN SENTENCE

Graph no.	Compactness	Randomness	Edge density
fig.(a)पीना _v ¹ , दूध _n ¹	0.75	0.66	0.70
fig.(b)पीना _v ² , दूध _n ¹	1.00	0.68	0.50
fig.(c)पीना _v ¹ , दूध _n ²	0.84	0.76	0.46
fig.(d)पीना _v ² , दूध _n ²	1.04	0.77	0.40

Here $p(v)$ is the vertex probability which is given by $\{\deg(v)/2|E|\}_{v \in V}$. Randomness of a graph have value in the range 0 to 1, where smaller value shows the least clustering solution. Graph randomness measure for a disconnected graph is always zero and for a fully connected graph is one. Graph (d) in Fig. 4 have highest randomness in Table II. This shows that nodes in graph (d) are more important as compare to other graphs and have high clustering solution.

a) Edge density

Edge density is also the graph clustering connectivity measure which is measured by calculating the ratio of number of edges of a complete graph with total number of vertices (given by $({}^{|V|}C_2)$).

$$ED(G) = |E(G)| / ({}^{|V|}C_2), \quad (12)$$

Where $ED(G)$ has a range [1,0], 1 for completely connected graph and 0 for totally disconnected graph [19]. A cluster having highest value of edge density is called important cluster. Here in Table II. edge density for graph (fig.(a)) is high.

In Table II we see that graph (a) produced the correct sense in given context (according to graph clustering based measures), which is produced by edge density. In sum after evaluating graph clustering based measures on so many sentences we can say that clustering based measures does not produce better results in comparison to node neighbor based measures.

V. RESULTS

The method proposed by authors in this paper was tested on a sample of 500 sentences of Hindi using Hindi WordNet as resource. For extracting exact sense of word in a given context, calculate centrality measures (based on node neighbors and graph clustering) for all open class words (ambiguous).

After so many experiments results indicate that node neighbor based connectivity measures produce better results than graph clustering based connectivity measures. The performance of Degree centrality measures are shown below in graph with respect to no. of incident edges to the correct sense and % of correctly classified words.

Node neighbor and graph clustering connectivity measures corresponding to मेहनत सफलता की कुंजी है। सफलता का फल मीठा होता है। उसने खाना खाया। are given below in table III and IV respectively. In this way, running the algorithm for different sentences for all open class words we get results as follows:

No of sentences = 500

No of Ambiguous words = 1200

No. of words correctly classified by node neighbor connectivity measures = 723

Accuracy obtained by node neighbor connectivity measures=60.25%

No. of words correctly classified by graph clustering measures=498

Accuracy obtained by graph clustering connectivity measures=41.25%

TABLE III. CONNECTIVITY MEASURES TABLE BASED ON NODE NEIGHBORS FOR AMBIGUOUS WORDS

Words	Degree	PR	KPP	HITS	Betweenness	Sentence
खाना _n ¹	0.14	0.64	0.21	0.34	0.01	उसने खाना खाया ।
खाना _n ³	0.09	0.43	0.18	0.22	0.01	
खाना _n ⁶	0.09	0.09	0.19	0.22	0.17	
खाना _v ¹	0.19	1.35	0.24	0.49	0.01	
खाना _v ²	0.19	1.37	0.28	0.40	0.02	
कुंजी _n ¹	0.25	0.64	0.63	0.37	0.12	मेहनत सफलता की कुंजी है ।
कुंजी _n ²	0.08	0.15	0.49	0.26	0.00	
कुंजी _n ³	0.16	0.43	0.42	0.32	0.16	
फल _n ¹	0.15	0.60	0.40	0.24	0.05	सफलता का फल मीठा होता है ।
फल _n ²	0.15	0.57	0.39	0.21	0.05	
फल _n ³	0.15	0.57	0.36	0.21	0.33	
कार्य _n ²	0.15	0.60	0.40	0.53	0.42	

कार्य _n ⁵	0.15	0.71	0.38	0.43	0.41	
---------------------------------	------	------	------	------	------	--

TABLE IV. CONNECTIVITY MEASURES TABLE BASED ON GRAPH CLUSTERING FOR AMBIGUOUS WORDS

Words	Compactness	Randomness	Edge density
खाना _n ¹ , खाना _v ¹	0.83	0.65	0.60
खाना _n ³ , खाना _v ¹	1.00	0.67	0.50
खाना _n ⁶ , खाना _v ¹	1.00	1.00	0.50
कुजी _n ¹ , मेहनत _n ¹	0.93	0.81	0.42
कुजी _n ² , मेहनत _n ¹	0.98	0.91	0.28
कुजी _n ³ , मेहनत _n ¹	0.91	1.01	0.20
फल _n ¹ , मेहनत _n ¹	1.04	0.83	0.38
फल _n ² , मेहनत _n ¹	1.00	0.89	0.32
फल _n ³ , मेहनत _n ¹	1.00	0.89	0.32

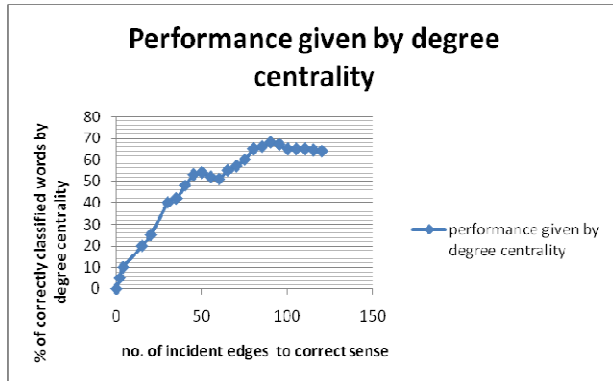


Fig. 5. Performance given by Degree Centrality

TABLE V. PERFORMANCE COMPARISON

Measures	Performance
Measures based on node neighbors	60.25%
Measures based on graph clustering	41.25%
Effect of context window size and stemming [17]	54.81%

VI. CONCLUSION

In this paper, authors proposed a graph based WSD algorithm for measuring context meaning for all open class words present in a sentence for Hindi language. Graph centrality measures results shows that measures based on node neighbors produce better result than the measures based on graph clustering. The method presented here is better than previous approaches of Hindi WSD because it works for all open class words (noun, verb, adverb, adjective) and disambiguates all the words present in the sentence simultaneously.

VII. REFERENCES

- [1] Hindi Wordnet from Center for Indian Language Technology Solutions, IIT Bombay India <http://www.cfilt.iitb.ac.in/wordnet/webhwn/>
- [2] Tanveer Siddiqui and U.S. Tiwari, "Natural language processing and information retrieval" 2010.
- [3] Robert Navigli, "Word Sense Disambiguation: A Survey", ACM Computing Surveys, Vol. 41, No. 2, Article 10, Publication date: February 2009.
- [4] TURING, A. M. 1950. Computing machinery and intelligence. Mind 54, 443–460.
- [5] Sandeep Kumar Vishwakarma, Chanchal Kumar Vishwakarma, "A Graph Based Approach to Word Sense Disambiguation for Hindi Language", International Journal of Scientific Research Engineering & Technology (IJSRET) Volume 1 Issue5 pp 313-318 August 2012.
- [6] RADU FLORIAN, SILVIU CUCERZAN, CHARLES SCHAFER and DAVID YAROWSKY, "Combining classifiers for word sense disambiguation", Natural Language Engineering 8 (4): 327–341. © 2002 Cambridge University Press.
- [7] CHAN, Y. S., NG, H. T., AND CHIANG, "Word sense disambiguation improves statistical machine translation", In Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics (Prague, Czech Republic). 33–40, 2007.
- [8] Ravi Sinha and Rada Mihalcea, "Unsupervised Graph-based Word Sense Disambiguation Using Measures of Word Semantic Similarity", IEEE 2007.
- [9] Roberto Navigli and Mirella Lapata, "An Experimental Study of Graph Connectivity for Unsupervised Word Sense Disambiguation", IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 32, NO. 4, APRIL 2010.
- [10] Manish Sinha, Mahesh Kumar Reddy, R Pushpak Bhattacharyya, Prabhakar Pandey and Laxmi Kashyap, "Hindi Word Sense Disambiguation", Indian Institute of Technology Bombay, Department of Computer Science and Engineering Mumbai, 2008.
- [11] Neetu Mishra, Shashi Yadav and Tanveer J. Siddiqui, "An Unsupervised Approach to Hindi Word Sense Disambiguation," M.Tech Thesis, Indian Institute of Information Technology, Allahabad. UP, India, 2009.
- [12] Rohan, "Word Sense Disambiguation For Hindi language", M.tech Thesis, Thapar University Patiyala, CSE Dept., India, 2007.
- [13] Avneet Kaur, "Development of an Approach for Disambiguating Ambiguous Hindi postposition," International Journal of Computer Applications (0975 – 8887), vol.5, no.9, August 2010.
- [14] Ravi Sinha and Rada Mihalcea, "Unsupervised Graph-based Word Sense Disambiguation Using Measures of Word Semantic Similarity," IEEE International Conference on Semantic Computing, pp. 363 – 369, Sept. 2007.
- [15] Siva Reddy, Abhilash Inumella, Rajeev Sangal, Soma Paul, "All Words Unsupervised Semantic Category Labeling for Hindi" Proceedings of the International Conference RANLP, Borovets, Bulgaria, pages 365–369, September 2009.
- [16] Parul Rastogi and Dr. S.K. Dwivedi, "Performance comparison of Word Sense Disambiguation (WSD) Algorithm on Hindi Language Supporting Search Engines", International Journal of Computer Science Issues, vol. 8, issue.2, March 2011.
- [17] Satyendr Singh and Tanveer J. Siddiqui, "Evaluating Effect of Context Window Size, Stemming and Stop Word Removal on Hindi Word Sense Disambiguation" 978-1-4673-1090-1/12/\$31.00 ©2012 IEEE.
- [18] Sabnam Kumari, Prof. Paramjit Singh, "Genetic Algorithm Based Hindi Word Sense Disambiguation", International Journal of Computer Science and Mobile Computing Vol.2 Issue. 5, May- 2013, pg. 139-144 ©2013.
- [19] Rada Mihalcea, Dragomir Ramdev, "Graph- Based Natural Language Processing", © 2011 Cambridge University Press.
- [20] R. Mihalcea, P. Tarau, and E. Figa, "Pagerank on Semantic Networks, with Application to Word Sense Disambiguation", Proc. 20th Int'l Conf. Computational Linguistics, 2004
- [21] J.M. Kleinberg, "Authoritative Sources in a Hyperlinked Environment", Proc. Ninth Symp. Discrete Algorithms, pp. 668-677, 1998.
- [22] S. Brin and M. Page, "Anatomy of a Large-Scale Hypertextual Web Search Engine", Proc. Seventh Conf. World Wide Web, pp. 107-117, 1998.