IAES International Journal of Artificial Intelligence (IJ-AI)
Vol. 13, No. 4, December 2024, pp. 4957~4964
ISSN: 2252-8938, DOI: 10.11591/ijai.v13.i4.pp4957-4964

Unsupervised hindi word sense disambiguation using graph-based centrality measures

Prajna Jha[1], Shreya Agarwal[1], Ali Abbas[1], Satyendr Singh[2], Tanveer Jahan Siddiqui[1]
[1]Department of Electronics and Communication, Faculty of Science, University of Allahabad, Prayagraj, India
[2]Department of Computer Science and Engineering, School of Engineering and Technology, BML Munjal University, Kapriwas, India

Article Info

Article history:

Received Jan 27, 2024
Revised Mar 27, 2024
Accepted Apr 17, 2024

Keywords:

Hindi WordNet
Natural language processing
Path-based similarity measure
Weighted graph-based centrality measures
Word sense disambiguation

ABSTRACT

The task of word sense disambiguation (WSD) plays a key role in multiple applications of natural language processing. In this paper, we propose a novel unsupervised method for targeted Hindi WSD task. First, we create a weighted graph where the nodes correspond to various synsets of the target word and the neighboring context words. The edges in the graph represent the semantic relations between these synsets in the Hindi WordNet hierarchy. A path-based similarity measure, namely Leacock-Chodorow similarity measure, is used to assign weights to edges. An unsupervised weighted graph-based centrality algorithm is used to identify the correct sense of a target word in a given context. The performance of the proposed algorithm is measured on 20 ambiguous Hindi nouns using four different graph-based centrality measures. We observed a maximum accuracy of 66.92% using PageRank centrality measure which is significantly better than earlier reported graph-based Hindi WSD algorithmsevaluated on the same dataset.

Corresponding Author:

Prajna Jha
Department of Electronics and Communication, Faculty of Science, University of Allahabad
Senate House, University Road, Old Katra, Prayagraj, Uttar Pradesh 211002, India
Email: pragya.jha.jk@gmail.com

1. INTRODUCTION

There is a growing demand for developing tools and technologies that can aid in exchanging and understanding ideas on a global scale. Natural language processing (NLP) is of great help in meeting these challenges by enabling machines to understand and generate human language. Machine translation, sentiment analysis, text summarization, speech recognition, and optical character recognition. Are some of the applications of NLP. However, there are certain challenges associated with these NLP applications. One of the key challenges is resolving the ambiguity present in language [1]. Ambiguity is an inherent problem to all human languages. It refers to the situation where a word, phrase, sentence, or entire discourse can be interpreted in more than one way. Ambiguity in a language occurs due to various reasons such as homonyms (words with similar spelling and pronunciation, but have different meanings), polysemous words (words with multiple meanings), or figurative (use of idioms and metaphors) in text or speech. The problem of ambiguity can also arise due to differences in perspective knowledge of contextual information. For example, in the following Hindi sentences, the word 'कलम' {*kalam*} corresponds to two different meanings:

"माली ने गुलाब की **कलम** तैयार की। {*Maali ne gulab ki kalam taiyyar ki*}
English translation: The gardener prepared the rose pen.

मैंने आज एक लाल **कलम** खरीदी।{*Maine aaj ek laal kalam khareedi*}
English translation: I bought a red pen today.

Here and henceforth, we will provide transliteration (in curly braces) and translation in English of Hindi text wherever it is needed for comprehending the meaning of the content. As it is evident from the context, in the second sentence, the word 'कलम' is used in the sense of 'an instrument used for writing something', whereas, in the first sentence, it is used in the sense of grafting. We as a human being can easily interpret the meaning of words but it poses a significant challenge to machines. The task of identifying the meaning of an ambiguous word automatically is known as word sense disambiguation (WSD). WSD is one of the key research areas in NLP. It is considered an AI-complete problem [1]. WSD is an intermediate task in NLP applications like machine translation while in applications like information retrieval, and text summarization. It can help in improving the performance.

The difficulty present in WSD originates from multiple factors such as the establishment of meaning of words; determination of the granularity of sense inventory, nature of words (domain-specific or unrestricted). The task of WSD has evolved over the decade incorporating new techniques. The earliest known method for WSD task is the dictionary-based Lesk's algorithm which uses direct overlap between the context and sense definitions to disambiguate an ambiguous word. The method was later modified to use standard lexical resources such as WordNet [2] and other similarity measures besides direct overlap similarity measures [3].

For the English language, many supervised algorithms have been explored and proposed by scientists worldwide. Most of these algorithms utilize lexical and contextual knowledge for disambiguation [1] and have been successful in achieving good accuracy. However, these methods require a sense-tagged corpus for training. The creation of such a corpus is a labor and time-consuming task. This poses a problem in applications of supervised approaches for resource-constraint languages like Hindi and other Indian languages. This motivates researchers to explore unsupervised methods for WSD. The unsupervised approach for disambiguating word sense was first introduced by Yarowsky [4], [5]. He used a small number of seed instances for different senses of a word and then iteratively tag instances in an untagged corpus using collocate. He proposed the use of one sense per discourse, and one sense per collocation to tag remaining instances [5]. The main problem that was observed with the unsupervised approach is that syntactic-semantic relations between word pairs and the conceptual information of nearby words were not clearly incorporated. The graph-based approach provides a powerful representation for modeling structural and complex relationships existing between data units. Some of the recent research applications with graph-based applications in the field of AI can be found in [6]. The graph-based approaches are being vastly used in the WSD research area [7]. In a graph-based algorithm, the lexical entities are represented as nodes and the syntactic-semantic relationship between them is represented as edges in a graph.

The graph-based algorithms are quite useful for low-resource languages as it does not require large training datasets. Owing to these properties and their successful applications in English WSD tasks, we propose and evaluate a novel graph-based algorithm for the Hindi WSD. The target words are ambiguous Hindi nouns. We use the Leacock-Chodorow method [8] to assign weights to edges and disambiguate a word by selecting a sense node that receives the highest score using the graph centrality measures. We investigate four different graph-based centrality measures: closeness, betweenness, eigenvector, and PageRank. Earlier works involving graph-based Hindi WSD use minimum cost spanning tree [9] or perform a random walk for disambiguation [10], whereas this study uses: i) Leacock Chodorow method, a path-based similarity measure, to compute the edge weight, and ii) utilizes scores assigned to nodes using graph centrality measures in disambiguation.

We experimentally evaluate the performance of the proposed algorithm using four different centrality measures and compare their performance on 20 polysemous Hindi nouns used in [11]. The dataset is a part of the sense annotated dataset available on the technology development for Indian languages (TDIL) website [12]. Singh *et al.* [11] uses the Leacock-Chodorow similarity score in a Lesk-like setting. The choice of the dataset makes it possible to compare the performance of the proposed algorithm. We obtain an overall average accuracy of 66.92% using the PageRank algorithm, followed by 66.49% obtained using the closeness centrality algorithm which is better than the accuracy reported in [10], [11].

The paper is divided into the following sections. In section 1, the problem of word ambiguity in the Hindi language and our approach to resolve it is briefly introduced; section 2 discusses existing graph-based algorithms applied for WSD in English and Hindi languages. In section 3, we introduce the graph-based centrality measures being used in this work. Section 4 provides the details of the proposed methodology. The experiment, evaluation, and comparative results followed by a detailed discussion of our experimental observations in section 5. In section 6, we conclude our paper and suggest future directions.

2. RELATED WORKS

This section briefly reviews some of the earlier graph-based WSD methods. One of the early works involving WordNet graph for WSD task is reported in [13]. In this paper, the minimum semantic distance between pairwise synsets thas been used to disambiguate ambiguous nouns appearing in a context window. The WordNet taxonomy and the notion of conceptual distance is used for resolving lexical ambiguities of nouns in [14]. An unsupervised knowledge-based WSD algorithm proposed in [15] uses PageRank on the graph extracted from the document for open-text WSD. The vertices of the graph were derived from synsets, and the edges were derived using semantic relations among WordNet synsets. Navigli [16] introduced the use of lexical chains and structural semantic interconnectionsfor disambiguation. He proposed a novel method of interlinking senses as a graph-based lexicon structure and assigning them to context words, and consecutively ranking them using the hyperlink-induced topic search (HITS) algorithm. Tsatsaronis *et al.* [17] implements four semantic graph representations on senseval-2 and senseval-3 datasets viz. spreading activation for network processing (SAN), Page Rank, HITS, and primitive rank (P-Rank).

Another work involving English WSD uses a co-occurrence graph in which vertices consist of words that occur together with the target word, and edges represent their frequency and identify sense by iteratively selecting highly connected hubs [18]. These hubs are treated as a depiction of the senses induced by the algorithm, the same way as clusters of examples in [19]. Agirre and Soroa [20] have shown unsupervised word sense induction (WSI) gives better results compared to supervised WSI in terms of F-score on SensEval-3 dataset [21]. Another way of applying the graph-based algorithm in unsupervised WSD is by exploiting the hierarchical property of the graph in the hierarchical random graph (HRG) algorithm [22]. This work uses a sense-tagged corpus similar to [20] and computes collocational weight by applying the Jaccard coefficient similarity on the target word and its context words. The result shows that HRGs outperform the Chinese whisper unweighted (CWU) baseline by 9.4%. A method of inducing sense using collocations in a graph is presented in [23]. The authors used Chinese Whispers and Jaccard similarity to populate the graph.

Narayanan and Bhattacharayya [24] formed a semantic directed-acyclic-graph (DAG) where vertices are the textual word synsets, and for each synset of the word, the link distance from the synset to the current vertex of the DAG and the current word from the text are determined using WordNet on SemCor-corpus [25]. An unsupervised English WSD algorithm proposed in [26] uses the centrality algorithms to the weighted graph and finds the most appropriate sense using the voting method from six different centrality measures. Some other notable works involving multilingual resources for disambiguation include [27], [28].

In recent years, efforts have been made by Indian researchers to explore the graph features, and graph measures for solving certain open-class research problems, WSD being one of them. The work involving Hindi WSD includes [9], [29]–[32]. According to Jain and Lobiyal [9], graph-based connectivity measures are used for Hindi WSD, the links between the synset nodes are created using various relationships defined in WordNet. However, no weights are assigned to these links and each link is assumed to take “unity” weight. The use of local and global graph connectivity measures was involved in [29]. A graph-based algorithm to disambiguate open-class words was proposed in [30] which uses node neighborhood connectivity measures for disambiguation. A novel idea that the association between words is governed by a gradual transition from being related to not related, i.e., there is a degree of fuzziness, was proposed by Jain and Lobiyal [31]. The authors developed a fuzzy Hindi WordNet and used it to perform Hindi WSD using fuzzy connectivity measures. The concept of cooperative game theory and fuzzy Hindi WordNet was used disambiguation in [32]. Research focusing on graph-based WSD is also applied in Indian languages such as Telugu [33]. The proposed work differs from all these earlier reported work as specified in section 1.

3. SEMANTIC SIMILARITY AND GRAPH-BASED CENTRALITY MEASURES

This section introduces the semantic similarity and graph centrality measures used in this study. Semantic similarity determines the likelihood estimation of the semantic association that exists between two semantic entities. It is usually obtained utilizing the information content of the manually annotated corpora or structured resources such as WordNet. A number of WordNet-based semantic similarity measures have been proposed to compute semantic similarity between the two synsets [8]. We use Leacock-Chodorow semantic similarity measure to determine the semantic association between synset pairs. The Leacock-Chodorow method considers the conceptual distance between the two concepts in the hierarchy/taxonomy of the lexical database such as WordNet. The similarity score between the two concepts is calculated by determining the shortest path containing the least common subsumer of two concepts, and its depth. Mathematically, the Leacock-Chodorow similarity measure can be expressed as (1):

$$LCS_{sim} = -\frac{\log(least\ common\ subsumer\ between\ c_i\ and\ c_{i+1})}{2*D} \quad (1)$$

Where c_i and c_{i+1} are the two concepts, and D is the maximum depth of the taxonomy. Here, for nouns, D is 12 [11] in the WordNet hierarchy. The graph-based centrality algorithm is applied to a graph or network to determine the significance of each node relative to other nodes in the graph. It calculates the centrality or importance of each node considering the location of the node in the graph, and its connectivity with the adjoining nodes. In our work, we have applied four centrality algorithms as following.

3.1. Closeness centrality

For a node v_i, closeness is defined as the reciprocal of the sum of shortest distance between v_0 to all the other nodes v_i over all the reachable nodes. For the weighted graph, the weight of the edges is considered in calculating the shortest path for a specific node. It can be expressed using (2).

$$Closeness\ (v_i) = \frac{n-1}{\sum_{v_{j=1}}^{n-1}(shortest\ path\ length\ (v_i, v_j, weight)} \quad (2)$$

3.2. Betweenness centrality

For a node v_i, betweenness is defined in terms of the total number of shortest paths existing between two node pairs v_x and v_y, and the number of such paths that passes through the intermediate node v_i. The process is repeated for all the adjoining pairs of nodes and node v_i in the graph. For weighted betweenness centrality, the edge weights are considered while computing the total weight of shortest paths between v_x and v_y, and the fraction of such weighted shortest paths that pass through the node v_i as in (3).

$$\mathrm{Betweenness\ (v_i)} = \sum_{\mathrm{v_x, v_y \in V}} \frac{\sigma(\mathrm{v_x, v_y | v_i})}{\sigma(\mathrm{v_x, v_y})} \quad (3)$$

where $\sigma(\mathrm{v_x, v_y})$ denotes the weighted shortest path existing between $\mathrm{v_x, v_y}$.

3.3. Eigenvector centrality

For a node v_i, its importance in the graph is dependent upon the importance of the neighboring nodes. Thus, in eigenvalue centrality of a node is directly proportional to the sum of the centrality score of its neighboring nodes. It is measured by calculating the principal eigenvector of the adjacency matrix. The process is iterated till convergence until the eigenvector centrality score for all the nodes is determined. The node of the graph forms the vector. Mathematically, eigenvector centrality for a node v_i which is the $\mathrm{i^{th}}$ entity of the vector x, can be expressed as in (4).

$$Eigenvector(v_i): Adj(G)x = \lambda x \quad (4)$$

where $Adj(G)$ is the adjacency matrix of the graph $G(V, E)$, and λ is the eigenvalue computed over the adjacency matrix. The largest value of the solution is selected for the eigenvalue.

3.4. PageRank

PageRank is used to assign scores or ranks to the nodes according to their relevance in the graph. It takes into account the linked structure of the graph. In the PageRank algorithm, each node linked to a given node casts its vote for that particular node. Initially, each node is given an equal score in the algorithm. Then the algorithm iteratively updates the score of the nodes until convergence is achieved. The nodes are ranked according to their score. Given that w_{ij} be the weight associated with the link connecting vertices v_i, v_j, w_{jk} be the weight for the link connecting v_j, v_k, PageRank is defined as (5).

$$PR(v_i) = (1-\alpha) + \alpha * \sum_{(v_i, v_j)\in E} \frac{w_{ki}}{\sum_{(v_k, v_j)\in E} w_{jk}} PR\left(v_j\right) \quad (5)$$

4. PROPOSED GRAPH-BASED METHOD FOR HINDI WORD SENSE DISAMBIGUATION

The proposed algorithm works by creating a weighted graph using the senses of target words and context words as vertices. The edges in the graph are created by joining a pair of vertices using the synset hierarchy of Hindi WordNet. The weight to an edge is assigned by computing semantic similarity between the pair of vertices. We use the Leacock-Chodorow similarity measure for computing semantic similarity between the two concepts. The algorithmic steps are detailed:

– Pre-processing: The dataset is pre-processed to remove stop words, to reduce morphological variants to their linguistic roots, and to assign part-of-speech tags to each word.
– Extraction of context window: The pre-processed data is used to extract a context window of size ±n words keeping the target word in the middle. In this work, a context window comprising of ±2 nouns with the target word in the middle is used.

$$\mathrm{CW} = \{\mathrm{w_{-n}, w_{-n-1}, \dots, w_{-1}, t_w, w_1 \dots, w_{n-1}, w_n}\}$$

- Graph creation: Extract a set of vertices, V, by extracting all the senses of the target word and all other words appearing in the context window from Hindi WordNet. Create an undirected weighted graph, $G = (V, E)$. The weight to an edge, e=(v_i, v_j) is assigned by computing semantic similarity between vertex v_i and v_j using the Leacock-Chodorow similarity measure.
- Computation of vertex score: The graph is traversed starting from the senses of the first word and each vertex is assigned a score using a graph-centrality measure as discussed in section 3. We experiment with four different measures–i) closeness, ii) betweenness, iii) eigenvector centrality, and iv) PageRank.
- Sense identification: The score of nodes representing the senses of the target word is extracted. The winner sense corresponds to the node with the highest score

An illustrative example of graph creation for the word 'उत्तर' {*uttar*} is shown in Figure 1. In the diagram, the vertices in the graph comprise of synsets of the target word and words appearing in ±2 context excluding stop words. These synsets are connected using WordNet-synset hierarchy. The weight of an edge is equal to the similarity score between its vertices obtained using the Leacock-Chodorow semantic similarity measure. The graph captures the syntactic-semantic relationship between the nodes, and thus, provides a powerful tool to perform WSD.

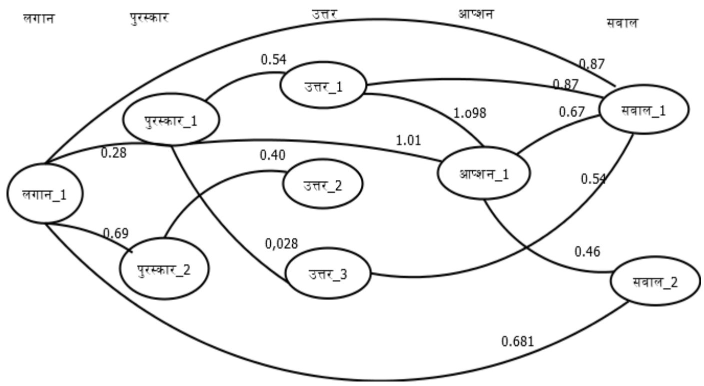

Figure 1. Weighted graph for the target word 'उत्तर' and context words

5. EXPERIMENT AND RESULT

We have performed our experiment on 20 ambiguous Hindi words taken from the sense-annotated Hindi dataset [12]. All these 20 ambiguous words are nouns. The list of target words used for the experiment and their corresponding sense counts are shown in Table 1. The dataset consists of sample instances for each sense of a polysemous word. The number of instances in the dataset is 965.

Table 1. Dataset description

#Senses	Target words (Nouns)
2	कोटा (*Quota/Kota*), हार (*Haar*), हल (*Hal*), सोना (*Gold*), विधि (*Vidhi*), माँग (*Maang*), दाम (*Daam*), तीर (*Teer*), तान (*Taan*), डाक (*Daak*), जेठ (*Jeth*), चंदा (*Chanda*), गुरु (*Guru*)
3	उत्तर (*Uttar*), कुंभ (*kumbh*), संबंध (*sambandh*), फल (*fal*), संक्रमण (*sankraman*), वचन (*vachan*)
4	मूल (*Mool*)

For disambiguation, a weighted-graph graph is created for each instance of the target word. Each node in the graph is then assigned a score using closeness, betweenness, eigenvector centrality algorithms, and PageRank. The process is repeated for each instance of all the 20 words. The accuracy of prediction for a particular word is obtained by averaging the accuracy of all the senses. The prediction accuracy of each of the target words in the dataset using closeness, betweenness, eigenvector, and PageRank measure is listed in Table 2.

From Table 2, it can be clearly observed that the PageRank and closeness measures perform significantly better than betweenness. The maximum accuracy of 0.6692 (averaged over all the instances) is obtained using PageRank, which is closely followed by closeness centrality (0.6649). The eigenvector centrality measure results in an overall accuracy of 0.6238. The worst performing case corresponds to the betweenness measure. The closeness centrality for the weighted graph considers the weight of links associated with the nodes to calculate the shortest path for each node pair. In certain cases, like sense-2 of 'कुंभ' ('*kumbh'*), out of 22 instances, closeness centrality accurately predicted the appropriate sense for 21 instances, followed by PageRank which predicted correctly 20 times. The reason for the poor performance of betweenness is that it assigns a score to each node by computing the maximum of the fraction of the shortest paths that passes through it. For disambiguation, we use the maximum scoring node among the senses of the target word. Due to the limited context two or more senses are assigned same score in which case the algorithm simply returns the sense listed first in the inventory.

Table 2. Prediction accuracy obtained using graph-based centrality measures

Target word (nouns)	Closeness	Betweenness	Eigenvector	PageRank
कोटा (*Quota/Kota*)	0.7181	0.0654	0.6278	0.6981
हार (*Haar*)	0.7944	0.5556	0.5833	0.7417
हल (*Hal*)	0.5952	0.4536	0.5595	0.5952
सोना (*Gold*)	0.6444	0.5989	0.5489	0.6722
विधि (*Vidhi*)	0.6528	0.5298	0.5651	0.6317
माँग (*Maang*)	0.6401	0.5795	0.6439	0.6420
दाम (*Daam*)	0.7958	0.2041	0.7334	0.7875
तीर (*Teer*)	0.6891	0.3309	0.6036	0.6945
तान (*Taan*)	0.7980	0.3101	0.6201	0.7980
डाक (*Daak*)	0.6785	0.4967	0.5519	0.6331
जेठ (*Jeth*)	0.55	0.6	0.45	0.65
चंदा (*Chanda*)	0.8132	0.3264	0.6544	0.7235
गुरु (*Guru*)	0.6645	0.4102	0.6538	0.6730
उत्तर (*Uttar*)	0.6354	0.5115	0.5926	0.6297
कुंभ (*kumbh*)	0.7174	0.4875	0.6008	0.7587
संबंध(*sambandh*)	0.7239	0.4257	0.6236	0.7
फल (*fal*)	0.6102	0.4031	0.6017	0.6973
संक्रमण (*sankraman*)	0.6991	0.4329	0.5253	0.5636
वचन (*vachan*)	0.7878	0.4873	0.6237	0.6721
मूल (*Mool*)	0.5725	0.4742	0.5570	0.6067
Average accuracy	0.6649	0.4851	0.6238	0.6692

Table 3 compares the performance of the proposed method with the baseline work reported in [11] which uses Leacok-Chodorow semantic similarity-based score in disambiguation and another more recent graph-based method reported in [10] that uses the same dataset. Jha *et al.* [10] uses semantic similarity to assign weights to edges and performs a random walk to achieve disambiguation. As evident from the table, all except the betweenness centrality measure performed better than the baseline [11]. Both PageRank and closeness measures performed better than the graph-based WSD method reported in [10] while the performance of Eigenvector centrality measure is comparable to it. We achieved a significant improvement of 10.33% over [11] and 5.57% over [10] using the PageRank measure. It should be noted that in [11] precision and recall measures are used for performance evaluation. The proposed algorithm provides an answer for each instance hence precision and recall are same as accuracy.

Table 3. Comparative analysis of proposed method over baseline

Method	Accuracy			
Proposed method	Closeness 0.6649	Betweenness 0.4851	Eigen Vector 0.6238	Page Rank 0.6692
Singh *et al.* [11]	0.6065			
Jha *et al.* [10]	0.6339			

6. CONCLUSION AND FUTURE WORK

In this paper, we propose and evaluate an unsupervised graph-based algorithm for the Hindi WSD task. We use the WordNet-based semantic similarity method to assign weight to edges. The disambiguation is done using scores assigned to the sense node. We experiment with four different centrality measures for node scoring. The PageRank measure with an overall accuracy of 66.92% performs better than all other measures. The closeness centrality measure performed quite close to PageRank with an observed accuracy of 66.49% over the entire dataset. The two best-performing cases of the proposed algorithm performed better than existing works on the same dataset. The improvement is quite significant. The proposed method using PageRank outperformed two of the earlier methods evaluated on the same dataset by a significant margin of 10.33% and 5.57%. The promising results obtained in this study demonstrate that unsupervised approaches can be effectively used for WSD tasks in the absence of tagged corpus and their performance can be significantly improved by enriching these approaches using knowledge from the existing lexical resources. Further investigations on larger datasets and involving other languages may be needed to confirm these findings. This work focuses on targeted WSD; however, it can be easily extended for all WSD. Future studies may explore this approach for other part of speech categories and for all WSD tasks involving Hindi, and other Indian languages.

ACKNOWLEDGEMENT

We express our gratitude to CFILT, IIT-Bombay for providing Hindi WordNet API as open source. We used this API for implementing parts of the proposed algorithm.

REFERENCES

[1] R. Navigli, “Word sense disambiguation: A survey,” *ACM Computing Surveys*, vol. 41, no. 2, 2009, doi: 10.1145/1459352.1459355.

[2] S. Banerjee and T. Pedersen, “An adapted lesk algorithm for word sense disambiguation using wordnet,” *Computational Linguistics and Intelligent Text Processing*, vol. 2276, pp. 136–145, 2002, doi: 10.1007/3-540-45715-1_11.

[3] S. Patwardhan, S. Banerjee, and T. Pedersen, “Using measures of semantic relatedness for word sense disambiguation,” *Computational Linguistics and Intelligent Text Processing*, vol. 2588, pp. 241–257, 2003, doi: 10.1007/3-540-36456-0_24.

[4] D. Yarowsky, “One sense per collocation,” *Human Language Technology: Proceedings of a Workshop Held*, Plainsboro, New Jersey, pp. 266-271, 1993, doi: 10.3115/1075671.1075731.

[5] D. Yarowsky, “Unsupervised word sense disambiguation rivaling supervised methods,” *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, vol. 1995, pp. 189–196, 1995, doi: 10.3115/981658.981684.

[6] N. Khan, Z. Ma, A. Ullah, and K. Polat, “Categorization of knowledge graph based recommendation methods and benchmark datasets from the perspectives of application scenarios: A comprehensive survey,” *Expert Systems with Applications*, vol. 206, 2022, doi: 10.1016/j.eswa.2022.117737.

[7] R. Mihalcea, “Unsupervised large-vocabularyword sense disambiguation with graph-based algorithms for sequence data labeling,” *HLT/EMNLP 2005 - Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, pp. 411–418, 2005, doi: 10.3115/1220575.1220627.

[8] S. Bhingardive, H. Redkar, P. Sappadla, D. Singh, and P. Bhattacharyya, “IndoWordNet: Similarity computing semantic similarity and relatedness using IndoWordNet,” *Proceedings of the 8th Global WordNet Conference, GWC 2016*, pp. 39–43, 2016.

[9] A. Jain and D. K. Lobiyal, “A new approach for unsupervised word sense disambiguation in Hindi language using graph connectivity measures,” *International Journal of Artificial Intelligence and Soft Computing*, vol. 4, no. 4, 2014, doi: 10.1504/ijaisc.2014.065800.

[10] P. Jha, S. Agarwal, A. Abbas, and T. J. Siddiqui, “A novel unsupervısed graph-based algorıthm for Hindi word sense disambiguation,” *SN Computer Science*, vol. 4, no. 5, 2023, doi: 10.1007/s42979-023-02116-1.

[11] S. Singh, V. K. Singh, and T. J. Siddiqui, “Hindi word sense disambiguation using semantic relatedness measure,” *Multi-disciplinary Trends in Artificial Intelligence: 7th International Workshop, MIWAI 2013*, Krabi, Thailand, vol. 8271, pp. 247–256, 2013, doi: 10.1007/978-3-642-44949-9_23.

[12] “Sense annotated Hindi corpus: Indian language technology proliferation and deployment centre,” *TDIL*. [Online]. Available: https://tdil-dc.in/index.php.

[13] M. Sussna, “Word sense disambiguation for free-text indexing using a massive semantic network,” *Proceedings of the Second International Conference on Information and Knowledge Management*, pp. 67–74, 1993, doi: 10.1145/170088.170106.

[14] E. Agirre and G. Rigau, “Word sense disambiguation using conceptual density,” *Proceedings of the 16th conference on Computational linguistics*, pp. 16-22, 1996, doi: 10.3115/992628.992635.

[15] R. Mihalcea, P. Tarau, and E. Figa, “PageRank on semantic networks, with application to word sense disambiguation,” *COLING 2004 - Proceedings of the 20th International Conference on Computational Linguistics*, 2004, doi: 10.3115/1220355.1220517.

[16] R. Navigli, “A structural approach to the automatic adjudication of word sense disagreements,” *Natural Language Engineering*, vol. 14, no. 4, pp. 547–573, 2008, doi: 10.1017/S1351324908004749.

[17] G. Tsatsaronis, I. Varlamis, and K. Nørvåg, “An experimental study on unsupervised graph-based word sense disambiguation,” *Computational Linguistics and Intelligent Text Processing*, vol. 6008, pp. 184–198, 2010, doi: 10.1007/978-3-642-12116-6_16.

[18] J. Véronis, “HyperLex: Lexical cartography for information retrieval,” *Computer Speech and Language*, vol. 18, no. 3, pp. 223–252, 2004, doi: 10.1016/j.csl.2004.05.002.

[19] A. Purandare and T. Pedersen, “Word sense discrimination by clustering contexts in vector and similarity spaces,” *The 8th Conference on Computational Natural Language Learning, CoNLL 2004 - Held in cooperation with HLT-NAACL 2004*, pp. 41–48, 2004.

[20] E. Agirre and A. Soroa, “Semeval-2007 task 02: Evaluating word sense induction and discrimination systems,” *ACL 2007 - SemEval 2007 - Proceedings of the 4th International Workshop on Semantic Evaluations*, vol. 2007, pp. 7–12, 2007.

[21] B. Synder and M. Palmer, “The English all-words task,” *Proceedings of SENSEVAL-3, the Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, pp. 41–43, 2004.

[22] I. P. Klapaftis and S. Manandhar, “Word sense induction and disambiguation using hierarchical random graphs,” *EMNLP 2010 - Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, pp. 745–755, 2010.

[23] I. P. Klapaftis and S. Manandhar, “Word sense induction using graphs of collocations,” *Frontiers in Artificial Intelligence and Applications*, vol. 178, pp. 298–302, 2008, doi: 10.3233/978-1-58603-891-5-298.

[24] U. E. Narayanan and P. Bhattacharayya, “Word sense disambiguation using semantic graph,” *Proceedings of the 1st International Global WordNet Conference,* 2002.

[25] G. A. Miller, C. Leacock, R. Tengi, and R. T. Bunker, “A semantic concordance,” *Proceedings of the Workshop on Human Languge Technology*, 1993, doi: 10.3115/1075671.1075742.

[26] R. Sinha and R. Mihalcea, “Unsupervised graph-based word sense disambiguation using measures of word semantic similarity,” *ICSC 2007 International Conference on Semantic Computing*, pp. 363–369, 2007, doi: 10.1109/ICSC.2007.87.

[27] E. Agirre and A. Soroa, “Using the multilingual central repository for graph-based word sense disambiguation,” *Proceedings of the 6th International Conference on Language Resources and Evaluation, LREC 2008*, pp. 1388–1392, 2008.

[28] C. Silberer and S. P. Ponzetto, “UHD: Cross-lingual word sense disambiguation using multilingual Co-occurrence graphs,” *ACL 2010 - SemEval 2010 - 5th International Workshop on Semantic Evaluation, Proceedings*, pp. 134–137, 2010.

[29] S. K. Vishwakarma and C. K. Vishwakarma, “A graph-based approach to word sense disambiguation for Hindi language,” *International Journal of Scientific Research Engineering & Technology*, vol. 1, pp. 313–318, 2012.

[30] A. Jain, S. Yadav, and D. Tayal, “Measuring context-meaning for open class words in Hindi language,” *2013 6th International Conference on Contemporary Computing, IC3 2013*, pp. 118–123, 2013, doi: 10.1109/IC3.2013.6612174.

[31] A. Jain and D. K. Lobiyal, “Fuzzy Hindi wordnet and word sense disambiguation using fuzzy graph connectivity measures,” *ACM Transactions on Asian and Low-Resource Language Information Processing*, vol. 15, no. 2, 2015, doi: 10.1145/2790079.

[32] G. Jain and D. K. Lobiyal, “Word sense disambiguation using cooperative game theory and fuzzy hindi wordnet based on ConceptNet,” *ACM Transactions on Asian and Low-Resource Language Information Processing*, vol. 21, no. 4, 2022, doi: 10.1145/3502739.

[33] N. Koppula, B. P. Rani, and K. S. Rao, “Word sense disambiguation in telugu language using knowledge-based approach,” *Advances in Intelligent Systems and Computing*, vol. 1090, pp. 153–161, 2020, doi: 10.1007/978-981-15-1480-7_13.

BIOGRAPHIES OF AUTHORS

Prajna Jha is a research scholar in the field of natural language processing and artificial intelligence in the Department of Electronics and Communication, University of Allahabad. She has completed Master of Technology (M.Tech.) in Computer Science from BanasthaliVidyapith, Tonk, Rajasthan. She has also worked at IIIT-Hyderabad on the Machine Translation Tool-Anusaaraka. She has published renowned articles in well-reputed conference, and journal on the topic of reordering, and word sense disambiguation in Hindi Language. She can be contacted at email: pragya.jha.jk@gmail.com or prajna_jha@allduniv.ac.in.

ShreyaAgarwal is a research scholar in the Department of Electronics and Communications in the field of NLP, in University of Allahabad. She did her masters in Computer Science and was awarded a Gold Medal from University of Allahabad. Her research interests include discourse analysis, natural language processing, generative AI, information extraction, and machine learning. She can be contacted at email: agarwal.shreya1994@gmail.com.

Ali Abbas has finished Master of Technology (M.Tech.) degree in Computer Technology, specializing in natural language processing. His research background encompasses over six years of experience in the field of document classification. He can be contacted at email: aliabbas367@gmail.com.

Dr. Satyendr Singh is working in the Department of Computer Science and Engineering at BML Munjal University, Gurugram. He obtained Ph.D. in Computer Science from the University of Allahabad, Prayagraj, India in 2015. His research interests include natural language processing, machine learning, and computational statistics. He can be contacted at email: satyendr@gmail.com.

Prof. Tanveer Jahan Siddiqui is Professor in computer science, Department of Electronics and Communication with more than 25 years of experience in teaching and research. She published more than 40 research articles in journals and conference proceeding, edited/authored books. She can be contacted at email: siddiqui.tanveer@gmail.com.