SN Computer Science (2023) 4:675
https://doi.org/10.1007/s42979-023-02116-1

ORIGINAL RESEARCH

A Novel Unsupervised Graph-Based Algorithm for Hindi Word Sense Disambiguation

Prajna Jha[1] · Shreya Agarwal[1] · Ali Abbas[1] · Tanveer J. Siddiqui[1]

Received: 7 March 2023 / Accepted: 3 July 2023

Abstract

Natural languages are inherently ambiguous. Ambiguities exist at many levels, word sense ambiguity being one of them. Resolving sense ambiguity is crucial in many Natural Language Processing applications. In this paper, we focus on word sense ambiguity and propose an unsupervised graph-based algorithm for Hindi Word Sense disambiguation task. The work is motivated by the encouraging results achieved by graph-based WSD algorithms for English and other European languages and the lack of wide-coverage sense annotated dataset for Hindi. The proposed algorithm creates a weighted graph wherein the nodes represent the senses of words appearing in the context of an ambiguous word and the edges depict relations between them. It uses semantic similarity derived from Hindi WordNet to assign weight to edges and a random walk-type algorithm to assign the most appropriate sense to a polysemous word in a given context. The evaluation has been done on a sense annotated dataset comprising 20 polysemous nouns. We observed an overall accuracy of 63.39% which is better than earlier reported work on the same dataset.

Keywords Word sense disambiguation · Unsupervised disambiguation · Graph-based word sense disambiguation · Hindi WSD

Introduction

Natural languages are inherently ambiguous. Ambiguity exists at multiple levels, polysemy is one of them. A polysemous word is a word with multiple meanings (senses). For example, 'हार' (Haar) as Noun has three senses defined in Hindi WordNet [9] (Fig. 1). The first sense pertain to 'defeat' sense of हार' (Haar). The second correspond to "string of flower (garland), pearl etc." and the third sense refers to "a neckpiece made of gold, silver, diamond, etc." The English translation is also provided in the Fig. 1 for improved readability.

In a given context, only one of these three senses will apply. For the sentence given in Fig. 2, it is clear from the context that 'हार' (Haar) is being used in the 'defeat' sense (Sense 1).

Human being can easily apply the contextually appropriate meaning of a polysemous word in a given sentence. But automatic identification of the correct sense of a word is quite difficult. Nevertheless, it is needed as an intermediary task in many applications of Natural Language Processing (NLP), such as machine translation, question answering and language understanding, and has potential to improve the performance many others like information retrieval [28], text summarization [14], disambiguating software requirements [29], etc. The process of selecting the most relevant meaning for a polysemous word in a given context automatically is known as Word Sense Disambiguation (WSD). A lot of work has already been done in WSD. However, majority of these works focus on English and other foreign Languages. Research involving Indian languages is still at its naive stage.

This article is part of the topical collection "Research Trends in Computational Intelligence" guest edited by Anshul Verma, Pradeepika Verma, Vivek Kumar Singh and S. Karthikeyan.

✉ Prajna Jha
pragya.jha.jk@gmail.com

Shreya Agarwal
agarwal.shreya1994@gmail.com

Ali Abbas
aliabbas367@gmail.com

Tanveer J. Siddiqui
siddiqui.tanveer@gmail.com

[1] Department of Electronics and Communication, University of Allahabad, Prayagraj, India

Published online: 02 September 2023

Fig. 1 Synsets of 'हार' (Haar)

Sense 1. हार, पराजय, असफलता, पराभव, शकिस्त, मात, अजय, अजै, अनभभिव, आपजय, वघिात, अभभिव, अभभिूत, अभषिंग, अभषिङ्ग, प्रसाह, अवगणन, अवजय, अवज्ञा, अवसाद, आवर्जन, भंग, भङ्ग, परभिाव, परीभाव: पराजति होने की अवस्था या भाव
"इस चुनाव में उसकी हार नश्चिति है । चुनाव में उसको पराजय हाथ लगी ।"

Sense 2. माला, हार, मालकिा, माल, अवतंस, अवतन्स, माल्यक:मनका, फूल आदि को सूत आदि में गोलाकार परिोकर बनाई हुई कोई वस्तु जो गले में पहनी जाती है
"उसके गले में मोतयिों की माला सुशोभति हो रही थी ।"

Sense 3. **हार, नेकलेस:**गले में पहनने का एक प्रकार का सोने, चाँदी आदि का गहना
"उसने हीरे का हार पहन रखा है ।"

English Translation:
Sense 1. Defeat , Paribhava: The state or condition of being defeated
"His defeat is certain in this election. He was defeated in the election."

Sense 2. Garland, Haar, Malika, Mala, Avatans, Avatans, Malayak: An object made by threading beads, flowers, etc. in a circular fashion, which is worn around the neck.
"A garland of pearls was adorning his neck."

Sense 3. Necklace: A type of gold, silver, etc. jewelry worn around the neck.
"She is wearing a diamond necklace."

Fig. 2 Example sentence for 'हार' (Haar)

रनों के लहिाज से यह वनडे इतहिास की सबसे बड़ी **हार** है।
Transliteration: { ranon ke lihaz se yeh oneday itihaas ki sabse badi haar hai }

English Translation: In terms of runs, this is the biggest defeat in ODI history.

Most of the early works in WSD are knowledge-based including the earliest known Lesk's algorithm. The development of wide-coverage lexical resources, sense inventories and sense annotated dataset, mostly for English, has contributed a lot in knowledge-based WSD approaches. However, these resources are language-specific and their development requires extensive manual effort. Further, they do not contain enough semantic knowledge to support advance concept-based NLP applications [11].

In order to overcome these limitations, machine learning approaches have been proposed which rely on text corpuses for knowledge needed for disambiguation. The best-known machine learning WSD algorithms existing to date are mostly supervised. But they require large amount of sense tagged data to achieve good results. Unfortunately, tagged dataset is not available for most of the languages in the world and creating sense tagged corpus manually is a quiet time-consuming and labour-intensive task. This poses major hindrance in its use for Hindi and other Indian languages which are deprived of such resources. Researchers have tried to handle the knowledge acquisition bottleneck by proposing automatic methods for creating new and extending the existing knowledge sources. The work reported in [11] involves development of a large-scale semantically enriched knowledge base, called KnowNet, using topic signatures acquired from the Web. The authors obtained better performance than any knowledge resources obtained manually or automatically by combining WordNet, Extended WordNet and Knowledge-NET. Ponzeto and Navigli [21] tried to overcome the knowledge bottleneck

by extending WordNet using semantic relations automatically extracted from Wikipedia. The extension results in a performance comparable to the state-of-the-art supervised approaches on open-text WSD.

A number of other researchers tried to overcome these limitations by enriching unsupervised approaches using the knowledge from existing lexical resources [6, 18]. These algorithms attempt to boost their performance by exploiting structure and relations existing in lexical resources. In this paper, we will refer these algorithms as unsupervised knowledge-based WSD methods. Like unsupervised WSD algorithms, these algorithms do not require labelled data for training but unlike unsupervised approaches which only discriminate amongst various senses. These algorithms are able to disambiguate instances with the help of an existing sense inventory. Given that lexical resources, such as WordNet have been developed already for Hindi [9], and other Indian languages, unsupervised knowledge-based methods offer a viable alternative. Unsupervised knowledge-based algorithms can be either graph-based ones or similarity-based. Earlier studies conducted on English language indicate that graph-based algorithms perform significantly better than similarity-based ones [17].

Motivated by the encouraging results achieved by graph-based WSD algorithms, we propose and evaluate a novel graph-based unsupervised method for Hindi WSD. The work presented here focuses on targeted word sense disambiguation task. Earlier work involving graph-based algorithm for WSD includes [7, 10, 16, 18]. Unsupervised graph-based algorithms are less investigated for Hindi. Mishra et al. [13], Jain and Lobiyal [1] are amongst the few investigations involving Unsupervised Hindi WSD. The work reported in this paper exploits the knowledge from Hindi WordNet in order to perform disambiguation using a novel unsupervised graph-based algorithm. We use the semantic information derived from Hindi WordNet to create a weighted graph. The algorithm uses a random walk type algorithm to assign the most appropriate sense to a polysemous word in a given context. To the best of our knowledge, no such evaluation has been done so far for Hindi WSD. For the purpose of evaluation, we have used a sense annotated dataset comprising 20 polysemous nouns. We observed an overall accuracy of 63.39% averaged over 20 words which is better than the similarity-based algorithm presented in [25] which also uses semantic similarity measures derived from Hindi WordNet and does the evaluation on the same dataset. But unlike the present work, they used a similarity-based approach for disambiguation. The dataset used in evaluation is derived from a publicly available corpus [27]. The rest of the paper is organised as follows:

In the next section, we briefly discuss the related work. The proposed methodology has been discussed in "Proposed methodology". Section "Performance Evaluation" presents the details of the dataset and the experimental investigation. Finally, conclusions are made in "Conclusion".

Related Work

Most of the early works in WSD were knowledge-based which try to identify the correct sense of a word based on the similarity between sense definitions in a sense inventory and some representation of the context. The main limitation comes from the need of language-specific knowledge resources and the limited context overlap. Subsequently, corpus-based approaches—both supervised as well as unsupervised—were proposed. This was possible partly due to the availability of online text corpuses in various languages. Supervised WSD approaches consider the task of disambiguation as a classification problem where each sense represents a category. The disambiguation problem is thus reduced to assigning each occurrence of an ambiguous word one of its sense category. In order to achieve this, a classifier is learned on a sense tagged corpus. Creating such a corpus involves manual effort and is language and domain specific. This limits its application to only those languages and domains for which sense tagged corpus is available. Unsupervised approaches do not require tagged corpus and therefore, can be applied easily to resource poor languages. However, purely unsupervised approach can only discriminate amongst various senses without assigning any tag.

The unsupervised approach for disambiguating word sense was first introduced by Yarowsky [5] which starts with a small number of manually provided seed collocation representatives of each sense of a word and tags instances on the basis of presence of these collocates in nearby context. The algorithm then automatically extracts additional collocation representatives of each sense from newly tagged instances and iteratively tags additional instances. In addition, she proposed the use of one sense per discourse, and one sense per collocation to tag remaining instances. Mihalcea [17] applied a graph-based sequence data labelling algorithm for WSD that exploits label dependencies for annotating sequences. She obtained an overall accuracy of 54.2% on standard SEMCOR corpus which was significantly better than previously proposed unsupervised WSD algorithms. In [18], an unsupervised graph-based method has been experimentally investigated using six different measures of word semantic similarity and several centrality algorithms. The results indicate that the right combination of similarity metric and graph centrality algorithms yields a performance at par with the state-of-the-art in unsupervised WSD methods on standard datasets. Agirre and Soroa [6] proposed a two-stage graph clustering approach which first creates a context similarity matrix using co-occurrence graph, prunes it to get an associated graph, and then performs a random walk

type algorithm to cluster it. Their algorithm outperforms all other unsupervised systems in Task 2 of Semeval-2007 competition. Agirre et al. [8] developed a WSD algorithm that uses PageRank and Personalized PageRank algorithm to perform random walk over the graph build from WordNet and Extended WordNet. The performance of their algorithm on English and Spanish datasets was comparable to the then-existing state-of-the-art methods. A huge amount of research exists in WSD area involving English and other European languages. Interested reader can find a detailed survey of existing approaches and applications of WSD in [12, 15]. Some of the notable works involving Hindi WSD include [1, 2, 23–25].

Following the development of IndoWordNet, several knowledge-based WSD methods have been proposed. In [20], the authors highlight the role of IndoWordNet for WSD task and present two unsupervised methods for WSD for Indian languages which use its semantic features and linked property to perform disambiguation. In [26], the authors used local and global graph connectivity measures in their work and obtained an accuracy of 65.17% on a sample Hindi corpus. Jain et al. [4] applied graph-based algorithm to disambiguate open class words. The graph was constructed for 500 sentences and Hindi WordNet with senses as vertices, and edges of syntactic relations between senses. The importance of vertices was established using connectivity measures based on node neighbour and graph clustering. Based on experimental investigation, they conclude that measures based on node neighbours produce better result than the measures based on graph clustering.

Mishra et al. [13] proposed an unsupervised approach for Hindi WSD which learns decision tree automatically from untagged instances using manually provided seed instances. Singh & Siddiqui [23] investigated the effect of stop word elimination and stemming on WSD. The experimental investigation reveals that these pre-processing steps contribute positively to Hindi WSD task in a Lesk-like setting. In [25], the authors used semantic similarity measure computed using WordNet hierarchy to disambiguate polysemous nouns. They obtained an overall average accuracy of 60.65% on a sense annotated dataset comprising of 20 polysemous Hindi nouns. Jain and Lobiyal presented an unsupervised graph-based approach for Hindi WSD [1]. The algorithm works on a sentence by sentence basis and disambiguates all the words appearing in a sentence simultaneously. A semantic graph is created for each interpretation of a given sentence with the help of Hindi WordNet. In the semantic graph, nodes represent word senses and edges represent relation between nodes. The graph with minimum cost of its spanning tree is used to provide the correct interpretation of the sentence. In [24], a supervised approach for Hindi WSD has been investigated. The work investigates Naïve Bayes classifier using 11 different features. The maximum accuracy was obtained when only the root form of the nouns was used as feature. In [22], three different algorithms, namely corpus-based Lesk, WSD using Co-occurrence and Classification Information model, were investigated for Hindi WSD. The authors experimented with different weighting schemes for computing overlap between context vector and sense definition. The corpus-based Lesk using tf-idf outperforms all others on a sense annotated dataset comprising of 60 nouns. The work reported in [2] presented a novel idea that association between words is governed by gradual transition from being related to not related, i.e. there is a degree of fuzziness. The authors generalised the relations defined in Hindi WordNet by assigning a membership value between [0, 1]. They also proposed fuzzy graph connectivity measures, and obtained significant improvement of 8% on Fuzzy Hindi WordNet when tested against sense tagged standard Hindi dataset. In [30], an improved performance is achieved by combining the concepts of Fuzzy Hindi WordNet and fuzzy semantic relations. Jain and Lobiyal [3] proposed an unsupervised WSD method for Hindi. They used a graph-based approach in which a sentence graph is created for each sentence. The sentence graph represents all possible interpretations in a sentence. From the sentence graph, sub-graphs are derived for all possible interpretations and network agglomeration is computed for each. The disambiguation is achieved by choosing the interpretation with maximum value of network agglomeration.

Proposed Methodology

In this section, we present the description of the graph-based model for word sense disambiguation for target words, and how the most appropriate sense for a target word in a given sentence is determined using random walk algorithm. The proposed algorithm considers the synsets of contextual words and computes the semantic relationship between the synsets. Using the similarity values, it creates a weighted graph and depends completely on the graph structure to ascertain the sense of the target word. The algorithm has three main parts:

(1) Computing shortest path similarity score for each word pairs.
(2) Creating a weighted graph G(V,E) from the set of synsets as vertices and semantic relationship between them as edges assigned with similarity scores, and
(3) Performing Random Walk on graph G(V,E).

For a given target word w, a context, W, is obtained for each instance as follows:

$$W = \left\{w_{-n}, w_{-n+1}, \ldots, w_{-2}, w_{-1}, \mathrm{w}, w_1, w_2, \ldots, w_{n-1}, w_n\right\}$$

For each $w_i \in W$, we obtain corresponding synsets as $D_{w_i} = \left\{S^1_{w_i}, S^2_{w_i}, S^3_{w_i}, \ldots, S^{N_{w_i}}_{w_i}\right\}$, where N_{w_i} is the total number of synsets for w_i derived from Hindi WordNet. For each consecutive synset pair, we compute the shortest path using hypernym hierarchy in WordNet. The semantic similarity between a vertex corresponding to jth sense of w_i to and vertex corresponding to kth sense of w_{i+1} is computed using the length of the shortest path between synsets as follows [20]:

$$Similarity = \frac{1}{\min\left(\text{pathe length}\left(S^j_{w_i}, S^k_{w_{i+1}}\right)\right)} \quad (1)$$

where, $w_i \in W, i = 1 \ldots N$.

In order to create a weighted graph, $G = (V, E)$ is created such that for every $synset \in D_{w_i}$, $i = 1, 2, \ldots, n$, there exist a vertex $v \in V$ in G. These vertices are linked together via directed edges to synsets of the word appearing next to it and a weight equal to the semantic similarity between them. The graph has to be completely connected. So we create an edge between those vertices also which have a 0.0 similarity value. The graph is to be completely connected for the algorithm to select the next node at random whilst performing random walk over the graph. The graph is created for each instance in the dataset separately.

The random walk algorithm starts from the initial vertex $v_0 \in V$. We use the first sense of the first word in the context window listed in Hindi WordNet. The number of steps the random walk takes determines the location of the next vertex $v\prime$ in graph G with reference to v_0. A path of 0 length will only include the initial vertex, a path of step 1 will include both initial vertex v_0, and one of its neighbouring node $v\prime$ selected at random, and so on. The algorithm stops when a vertex with no outgoing link is reached. The process is repeated with the initial vertex set to the next sense of the first context word. The path with maximum total edge weight (similarity) is used to predict the most appropriate sense to the target word. The algorithmic steps of the proposed methodology are summarised in "Algorithm" and an illustrative example is given in "An Illustration of the Proposed WSD Algorithm".

Algorithm

Algorithm of the proposed Graph Based WSD

1. Read an instance containing ambiguous target word
2. Remove stop words (Appendix I) and reduce morphological variants of words to their root form.
3. Extract context (CW) comprising of ±N words appearing in the context of target word w:

$$CW = \left\{w_{-N} \ldots w_{-2,} w_{-1}, w, w_1, w_2, \ldots, cw_N\right\}$$

V ←{ ϕ} and E ={ ϕ} where V is the set of vertices and E is the set of edges in G.

4. For each pair of adjacent words w_i and w_{i+1} in context window CW

4.1Let $Sw_i = \left\{S^{N_{w_i}}_{w_i} | N_{w_i} \text{ is the total number of synsets for } w_i\right\}$ and

$Sw_{i+1} = \left\{S^{N_{w_{i+1}}}_{w_{i+1}} | N_{w_{i+1}} \text{ is the total number of synsets for } w_{i+1}\right\}$ *be* set of all synsets of w_i and w_{i+1}respectively.

4.2 Update V as follows:

$$V \leftarrow V \cup Sw_i \cup Sw_{i+1}. \quad (2)$$

4.3 Compute shortest path between each synset in Sw_i and Sw_{i+1} using Hindi WordNet hypernym hierarchy

4.4 Create a weighted edge between each synset pair (Sw_i^j,Sw_{i+1}^k) where, Sw_i^j is j^{th} synset of word w_i (Sw_i^j) and synset k of word w_{i+1} (Sw_{i+1}^k) with the weight equal to the shortest path between them. Let E' be the set of all such edges.

4.5 Update E as: E ←E ∪ E' (3)

5. Perform a random walk on the semantic Graph G = {V, E} starting from each synset of the first context word leading to a set of solution sequences, S.

6. Find the sequence S' with maximum total edge weight. Label the target word with the sense predicted by sequence S'

An Illustration of the Proposed WSD Algorithm

Consider the following example sentence containing ambiguous word 'जेठ' (Jeth) which is to be disambiguated:

चाहे पूस का महीना हो या जेठ का पोशाक–पोशाक है, और उसे पहनना चाहिए.

Transliteration: Chahe pus ka mahina ho ya jeth ka poshak–poshak hai aur use pahnna chahiye.

English Translation: Whether it is the month of Pus or of Jeth, dress is a dress, and it should be worn.

Hindi WordNet has two sense of "जेठ". The first sense (3535) corresponds to 'brother-in-law' (elder brother of husband) sense and the second sense (779) corresponds to third month of Hindu calendar.

For the example sentence, the context window of 'जेठ' containing root form of words appearing in ±2 window is:

CW = {पूस#nounमहीना#noun जेठ#noun पोशाक#noun पहन#verb}.

The sequence of synsets D_{w_i} = {{पूस#noun_782} {महीना#noun_1691, महीना#noun_1690, महीना#noun_7658, महीना#noun_11020}, {जेठ#noun_3535,जेठ#noun_779},{पोशाक#noun_492},{पहन#verb_3939, पहन#verb_38122}}.

The weighted directed graph G(V,E) built for this five word context window is shown in Fig. 3.

The maximum weight sequence obtained for the sample instance is:

पूस#noun_782->महीना#noun_169->जेठ#noun_779->पोशाक#noun_492->पहन#verb_3939

Predicted sense of 'जेठ' in this context is sense 2 (779) which is contextually correct.

Performance Evaluation

Dataset

The experiment has been performed on a dataset comprising 20 ambiguous nouns from sense annotated dataset which is freely accessible. We have evaluated the proposed algorithm on 20 ambiguous nouns used in the baseline [25]. Table 1 lists these words. The total number of instances evaluated by us is 965. The average number of instance per word and per sense is around 48.25 and 20.10 respectively. The choice of this dataset makes it possible to compare the proposed work with earlier existing work evaluated on the same dataset without duplicating the effort [25]. The baseline work uses an unsupervised similarity-based approach for disambiguation. The disambiguation is achieved by finding the word sense that leads to maximum similarity between the corresponding sense definition and the context. Instead of using keyword-based similarity, it uses semantic similarity derived from WordNet hierarchy to disambiguate a word.

Experiments and Results

The dataset is first pre-processed as discussed in Sect. 3.1. For tokenization and POS tagging, we have used POS tagger developed for Indian languages. The work reported in

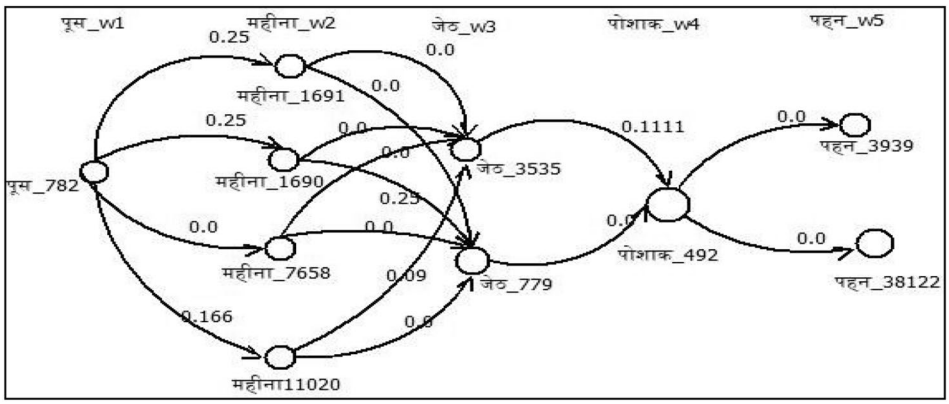

Fig. 3 Directed weighted graph G(V,E) for the example sentence

Table 1 Dataset description

#Senses	Target words (noun)
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)

Table 2 Accuracy (over 20 words)

Word	No. of sense	Total instances	Correctly predicted instances for each sense (y)	Accuracy of our algorithm (y/x)	Accuracy [25]
उत्तर	3	66 (#Is1-15; #Is2-35;#Is3-16)	40 (#Is1-11; #Is2-25; #Is3-7)	0.6060	0.9030
कुंभ	3	71 (#Is1-22; #Is2-22; #Is3-27)	45(#Is1-15; #Is2-16; #Is3-14)	**0.6338**	0.5701
कोटा	2	58 (#Is1-33; #Is2-25)	39 (#Is1-17, #Is2-22)	**0.6724**	0.6416
गुरु	2	44 (#Is1-26; #Is2-18)	32(#Is1-23, #Is2-9)	0.7272	0.6782
चंदा	2	54 (#Is1-20; #Is2-34)	40 (#Is1-19;#Is2-21)	**0.7407**	0.5687
जेठ	2	20 (#Is1-10; #Is2-10)	14 (#Is1-6, #Is2-8)	**0.7**	0.6666
डाक	2	36 (#Is1-14; #Is2-22)	25 (#Is1-5; #Is2-20)	**0.6944**	0.5000
तान	2	29 (#Is1-13; #Is2-16)	20 (#Is1-9; #Is2-8)	**0.5862**	0.4166
तीर	2	36 (#Is1-25; #Is2-11)	23 (#Is1-15; #Is2-8)	0.6388	0.7500
दाम	2	39 (#Is1-24; #Is2-15)	23 (#Is1-17; #Is2-6)	0.5897	0.7368
फल	3	50 (#Is1-23; #Is2-19; #Is3-8)	33 (#Is1-15; #Is2-10; #Is3-8)	**0.66**	0.6526
माँग	2	46 (#Is1-24; #Is2-22)	28 (#Is1-17; #Is2-11)	0.6087	0.6363
मूल	4	90 (#Is1-5; #Is2-35; #Is3-28; Is4-22)	63 (#Is1-3; #Is2-32; #Is3-20; #Is4-8)	0.7	0.9019
वचन	3	28 (#Is1-9; #Is2-11; #Is3-8)	19 (#Is1-8; #Is2-5; #Is3-6)	**0.6785**	0.3809
विधि	2	83 (#Is1-53; #Is2-30)	52 (#Is1-34; #Is2-18)	**0.6265**	0.5909
संक्रमण	3	56 (#Is1-14; #Is2-23; #Is3-19)	32(#Is1-7; #Is2-12; #Is3-13)	**0.5714**	0.1809
संबंध	3	34 (#Is1-14; #Is2-14; #Is3-6)	21 (#Is1-10; #Is2-8; #Is3-3)	**0.6176**	0.3888
सोना	2	44 (#Is1-26; #Is2-18)	26 (#Is1-16; #Is2-10)	0.5909	0.7500
हल	2	43 (#Is1-15; #Is2-28)	29 (#Is1-10; #Is2-19)	**0.6744**	0.6114
हार	2	38 (#Is1-20; #Is2-18)	25 (#Is1-9; #Is2-16)	**0.6578**	0.6052
Mean Accuracy				**0. 6339**	0.6065

Bold signifies the considerable improvement in accuracy we have achieved with our algorithm when compared with the results of the proposed work by authors Singh and Siddiui [25]

[25] forms the baseline. Test run is conducted for a fixed window size of 5($\pm$2 context window) as in [25]. Stop words are removed before extracting context window. All open class words (noun. verb, adjective, and adverb) are considered for creating context. For each instance, a connected graph is first created and then a random walk is performed on it to get solution sequences, each starting at a vertex corresponding to one of the synsets of the first word and terminating at a vertex corresponding to one of the synsets of the last word in the context. The sequence with maximum total edge weight is used to label the target word. For evaluation purpose, the average accuracy of over all the instances is computed for each of its senses. The total number of instances, the number of instances in each sense (Is_n), the number of correctly disambiguated instances and the observed accuracy for each of the 20 polysemous words are shown in Table 2. The table also compares the accuracy of each word with the accuracy reported in [25]. As can be seen from the Table 2, the overall average accuracy obtained using the proposed algorithm is 63.39% which is significantly better than the precision reported in [25]. The percentage improvement achieved over the baseline is 4.52%, which is calculated as follows:

$$\%\,\text{improvement} = \left(\frac{\text{improvement in accuracy}}{\text{original accuracy}}\right) \times 100 \quad (4)$$

The proposed algorithm answers each instance unlike [25], which do not consider unanswered instances in calculation. Considering this, the improvement is quite significant.

Discussion

As shown in Table 2, we achieved an accuracy of 63.39% averaged over all the instances of all the 20 words which is significantly better than the baseline. Unlike the work reported in [25] which uses nouns only in the context window, we have considered all POS categories. This helps in correct disambiguation in many cases. One important observation in the results obtained using the proposed random walk algorithm is that the performance is more consistent across the words. The minimum accuracy we observed is 0.57 for the word संक्रमण and the maximum

accuracy of 0.72 for the word चंदा, whereas the minimum accuracy and the maximum accuracy reported in [25] are 0.18 (for संक्रमण) and 0.9030 (for उत्तर). In most of the cases, the proposed algorithm exhibits better performance. Unlike [25], which first computes pair-wise similarity between all the senses of the context word and each sense of the target word and then uses maximum aggregated similarity for predicting appropriate sense, our algorithm requires semantic similarity only between senses of neighbouring words to assign weight to edges. This reduces computational complexity. Instead of using similarity, we use graph-based approach for disambiguation and perform a random walk on the directed weighted graph to get most appropriate sense tagged path sequence. In order to keep the graph connected, we have created edges for node pairs having 0.0 similarity also. A closer look of the poor-performing instances and also other instances reveals that there are cases which do not have common lexemes in their hypernym set, and hence the algorithm randomly follows one of the edges with 0.0 similarity which eventually leads to incorrect disambiguation resulting in a drop in average accuracy. For example, there are no common lexemes in the hypernym set of राजनयिक (Rajnayik—diplomatic) and संबंध (Sambandh-relation); क्षेत्र (kshetra) and संबंध (Sambandh). Although one can expect that presence of राजनयिक, क्षेत्र in the context of संबंध is a strong indicator of a particular sense. We feel that including other semantic relations in addition to hypernym may improve the situation. Unlike the algorithm in [25], the proposed algorithm can be used for all word sense disambiguations as well.

Conclusion

In this paper, we have proposed an unsupervised graph-based algorithm for Hindi Word Sense Disambiguation. The nodes in the graph are synsets of words appearing in a ± 2 context window of the ambiguous word. The weight to edges is assigned using semantic similarity between vertices derived from Hindi WordNet using hypernym hierarchy. The disambiguation is done by applying a random walk on the graph thus created. The proposed algorithm exhibits better performance than a variation of Lesk algorithm that uses semantic similarity score instead of direct overlap [25] when evaluated on the same set of words. The proposed algorithm does not require training data. It can be used for other Indian languages as well for which lexical resources like WordNet have already been developed. In the future, we would like to explore more relations to take care of null intersection between hypernym set of node pairs.

Appendix I

अगर	अगली	अगले	अच्छी	अति	अथवा	अधिक	अनुसार	अनेक
अन्य	अपना	अपनी	अपने	अब	अभी	अलावा	आ	आई
आएँ	आगे	आती	आदि	आने	आप	आम	आसपास	इतनी
इतने	इन	इनमे	इन्हीं	इन्हे	इस	इसका	इसकी	इसके
लिए	इसके	इसमें	इसलिए	इससे	इसी	इसीलिए	इसे	उतनी
उधर	उन	उनका	उनकी	उनके	उनमें	उनसे	उन्हीं	उन्हे
उन्हें	उन्होंने	उन्होने	उस	उसका	उसकी	उसके	उससे	उसी
उसे	ऊपर	एक	एक-एक	एवं	ऐसा	ऐसी	ऐसे	ओर
कई	कछ	कब	कभी	कभी-कभी	कम	कया	कर	करके
करता	करती	करते	करना	करनी	करने	करा	कराने	कराया
करेंगे	करेगा	करेगी	का	काफी	काफी	कि	किंतु	किए
कितनी	कितने	किन	किया	किये	किस	किसी	की	कुछ
कुल	के	कारण	कैसे	को	कोई	कौन	क्या	क्यो
क्योकि	गई	गईं	गए	गया	गयी	गये	चलता	चलने
चली	चाहती	चाहते	चाहिए	चाहे	चुका	चुकी	चुके	चुके
छह	छू	जगह	जब	जबकि	जल्द	जल्दी	जहाँ	जहां
जहां-तहां	जा	जाए	जाएं	जाएंगी	जाएगी	जाएँ	जाकर	जाता
जाती	जाते	जानना	जाना	जाने	जाये	जारी	जितना	जितनी
जिनमें	जिन्हें	जिसका	जिससे	जिसे	जी	जैसा	जैसे	जो
जोर	ज्यादा	ठीक	तक	तथा	तब	तभी	तरफ	तरह
तहत	ताकि	तीन	तो	तौर	था	थी	थे	थोडा
दरअसल	दिए	दिखाए	दिया	दी	दूर	दूसरी	दूसरे	दे

देंगी	देंगे	देकर	देता	देती	देते	देना	देने	दो
दोनो	द्वारा	न	नई	नए	नया	नहीं	नीचे	ने
पडता	पडने	पडा	पर	परंत	पहला	पहले	पांच	पाएं
पाँच	पीछे	पूरी	प्रति	प्रत्येक	फिर	बजाय	बजे	बडी
बढ़	बढ़ा	बढ़े	बताया	बन	बनाई	बनाए	बनाना	बनाने
बनी	बने	बल्कि	बहुत	बाकी	बाद	बार	बार-बार	बारे
बिना	बीच	बेहद	भी	मगर	मुताबिक	मे	में	यदि
यद्यपि	यह	यहाँ	यही	या	यानी	ये	रखना	रह
रहती	रही	रहे	रहेगा	रहेगी	रहो	रोका	लगभग	लगा
लगाई	लगे	लाकर	लाने	लिए	लिया	लिये	ली	ले
लेकर	लेकिन	लेगी	लेना	लेने	व	वनाट	वह	वहाँ
वहीं	वाला	वाली	वाले	वालो	विभिन्न	वे	वैसे	वो
शायद	सकता	सकती	सकते	सका	सके	सकेगा	सकेगी	सब
सबकी	सबके	सबसे	सभी	सहज	सही	सा	सात	साथ
साथ-साथ	साफ	सामने	सारे	सिर्फ	सीधे	से	हाँ	हम
हमने	हमारी	हमारे	हमें	हर	हां	हांलांकि	ही	हुआ
हुई	हुए	हूँ	है	हैं	हो	हों	होंगी	होगा
होगी	होता	होती	होते	होना	होनी	होने		

Funding No funding is available for the work reported in this paper.

Declarations

Conflict of interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

References

1. Jain A, Lobiyal DK. A new approach for unsupervised word sense disambiguation in Hindi language using graph connectivity measures. Int J Artif Intell Soft Comput. 2014;4(4):318–34.
2. Jain A, Lobiyal DK. Fuzzy Hindi WordNet and word sense disambiguation using fuzzy graph connectivity measures. ACM Trans Asian Low-Resour Lang Inf Process. 2015;15(2):1–31.
3. Jain A, Lobiyal DK. Unsupervised Hindi word sense disambiguation based on network agglomeration. In: 2015 2nd International Conference on Computing for Sustainable Global Development (INDIACom). 2015; 195–200.
4. Jain A, Yadav S, Tayal D. Measuring context-meaning for open class words in Hindi language. In:Proc. of 2013 Sixth International Conference on Contemporary Computing (IC3). IEEE. 2013;pp. 118–123.
5. Yarowsky D. Unsupervised word sense disambiguation rivaling supervised methods. In: 33rd annual meeting of the association for computational linguistics. Cambridge. 1995; pp. 189–196
6. Agirre E, Soroa A. Semeval-2007 Task 02: Evaluating word sense induction and discrimination systems. In: Proceedings of SemEval-2007, Prague. Czech Republic. 2007; pp. 7–12.
7. Agirre E, Martinez D, de Lacalle O, Soroa A. Two graph-based algorithms for state-of-the-art WSD. In: Proceedings of EMNLP-2006. Sydney, Australia; 2006, pp. 585–593.
8. Agirre E, de Lacalle OL, Soroa A. Random walks for knowledge-based word sense disambiguation. Comput Linguist. 2014;40(1):57–84.
9. https://www.cfilt.iitb.ac.in/wordnet/webhwn/
10. Klapaftis I, Manandhar S. Word sense induction using graphs of collocations. In: ECAI. July 2008. pp. 298–302. http://dx.doi.org/https://doi.org/10.3233/978-1-58603-891-5-298
11. Cuadros M, Rigau G. KnowNet: building a large net of knowledge from the Web. In Proc. of COLING-08.2008; pp161–168.
12. Bevilacqua M, Pasini T, Raganato A, Navigli R. Recent trends in word sense disambiguation: a survey. In: Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21. International Joint Conference on Artificial Intelligence, Inc. 2021; pp. 4330–4338.
13. Mishra N, Yadav S, Siddiqui TJ. An unsupervised approach to hindi word sense disambiguation. In: Tiwary US, Siddiqui TJ, Radhakrishna M, Tiwari MD (Eds) Proceedings of the First International Conference on Intelligent Human Computer Interaction. Springer, New Delhi. 2009. https://doi.org/10.1007/978-81-8489-203-1_32
14. Kouris P, Alexandridis G, Stafylopatis A. Abstractive text summarization: enhancing sequence-to-sequence models using word sense disambiguation and semantic content generalization. Comput Linguist. 2021;47(4):813–85.
15. Navigli R. Word sense disambiguation: a survey. ACM Comput Surv. 2009;41(2):1–69. https://doi.org/10.1145/1459352.1459355.
16. Mihalcea R, Tarau P, Figa E. Pagerank on semantic networks with application to word sense disambiguation. In: COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics. 2004; pp. 1126–1132.
17. Mihalcea R. Unsupervised large-vocabulary word sense disambiguation with graph-based algorithms for sequence data labeling. In: Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing. 2005;pp 411–418. DOI: https://doi.org/10.3115/1220575.1220627
18. Sinha R, Mihalcea R. Unsupervised graph-based word sense disambiguation using measures of semantic similarity. In the Proceedings of International Conference on Semantic Computing. IEEE. 2007; pp. 363–369. http://dx.doi.org/https://doi.org/10.1109/ICSC.2007.87.

19. Singh S, Siddiqui TJ. Role of karaka relations in hindi word sense disambiguation. J Inf Technol Res. 2015;8(3):21–42. https://doi.org/10.4018/JITR.2015070102.
20. Bhingardive S, Redkar H,Sappadla P, Singh D, and Bhattacharyya P. IndoWordNet::similarity-computing semanticsimilarity and relatedness using indoWordNet. In: Proceedings of the 8th Global WordNet Conference (GWC). 2016; pp. 39–43
21. Ponzetto SP, Navigli R. Knowledge-rich word sense disambiguation rivaling supervised systems. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics. Uppsala, Sweden: Association for Computational Linguistics. 2010; Pp. 1522–1531.
22. Singh S, Siddiqui TJ. Utilizing corpus statistics for hindi word sense disambiguation. Int Arab J Inform Technol. 2015;12(6A):755–63.
23. Singh S and Siddiqui Tanveer J. Evaluating effect of context window size, stemming and stop word removal on Hindi word sense disambiguation. In: International Conference on Information Retrieval & Knowledge Management (CAMP). 2012.
24. Singh S, SiddiquiTanveer J, Sharma Sunil K. Naïve Bayes classifier for Hindi word sense disambiguation. In: Proceedings of the 7th ACM India computing conference. 2014; pp. 1–8.
25. Singh S, Singh VK, Siddiqui TJ. Hindi word sense disambiguation using semantic relatedness measure. In: the Proceedings of MIWAI 2013, LNCS 8271, Springer. Berlin. 2013. pp. 247–256.
26. Vishwakarma SK, Vishkarma CK. A graph based approach to word sense disambiguation for Hindi language. Int J Sci Res Eng Technol (IJSRET). 2012;1:313–8.
27. Sense Annotated Hindi Corpus: Indian Language Technology Proliferation and Deployment Centre. https://tdil-dc.in/index.php
28. Zhong Z, Ng HT. Word sense disambiguation improves information retrieval. In: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics. Jeju, Republic of Korea. 2012, pp 273–282.
29. HussainMH, KhanumMA. Word sense disambiguation in software requirement specifications using wordnet and association mining rule. ICTCS '16: Proceedings of the Second International Conference on Information and Communication Technology for Competitive Strategies, March 2016, Article No.: 119, Pages 1–4.
30. Jain G, Lobiyal DK. Word sense disambiguation of hindi text using fuzzified semantic relations and fuzzy hindi WordNet. 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence), Noida, India, 2019, pp. 494–497.

Terms and Conditions

Springer Nature journal content, brought to you courtesy of Springer Nature Customer Service Center GmbH (“Springer Nature”).
Springer Nature supports a reasonable amount of sharing of research papers by authors, subscribers and authorised users (“Users”), for small-scale personal, non-commercial use provided that all copyright, trade and service marks and other proprietary notices are maintained. By accessing, sharing, receiving or otherwise using the Springer Nature journal content you agree to these terms of use (“Terms”). For these purposes, Springer Nature considers academic use (by researchers and students) to be non-commercial.
These Terms are supplementary and will apply in addition to any applicable website terms and conditions, a relevant site licence or a personal subscription. These Terms will prevail over any conflict or ambiguity with regards to the relevant terms, a site licence or a personal subscription (to the extent of the conflict or ambiguity only). For Creative Commons-licensed articles, the terms of the Creative Commons license used will apply.
We collect and use personal data to provide access to the Springer Nature journal content. We may also use these personal data internally within ResearchGate and Springer Nature and as agreed share it, in an anonymised way, for purposes of tracking, analysis and reporting. We will not otherwise disclose your personal data outside the ResearchGate or the Springer Nature group of companies unless we have your permission as detailed in the Privacy Policy.
While Users may use the Springer Nature journal content for small scale, personal non-commercial use, it is important to note that Users may not:

1. use such content for the purpose of providing other users with access on a regular or large scale basis or as a means to circumvent access control;
2. use such content where to do so would be considered a criminal or statutory offence in any jurisdiction, or gives rise to civil liability, or is otherwise unlawful;
3. falsely or misleadingly imply or suggest endorsement, approval , sponsorship, or association unless explicitly agreed to by Springer Nature in writing;
4. use bots or other automated methods to access the content or redirect messages
5. override any security feature or exclusionary protocol; or
6. share the content in order to create substitute for Springer Nature products or services or a systematic database of Springer Nature journal content.

In line with the restriction against commercial use, Springer Nature does not permit the creation of a product or service that creates revenue, royalties, rent or income from our content or its inclusion as part of a paid for service or for other commercial gain. Springer Nature journal content cannot be used for inter-library loans and librarians may not upload Springer Nature journal content on a large scale into their, or any other, institutional repository.
These terms of use are reviewed regularly and may be amended at any time. Springer Nature is not obligated to publish any information or content on this website and may remove it or features or functionality at our sole discretion, at any time with or without notice. Springer Nature may revoke this licence to you at any time and remove access to any copies of the Springer Nature journal content which have been saved.
To the fullest extent permitted by law, Springer Nature makes no warranties, representations or guarantees to Users, either express or implied with respect to the Springer nature journal content and all parties disclaim and waive any implied warranties or warranties imposed by law, including merchantability or fitness for any particular purpose.
Please note that these rights do not automatically extend to content, data or other material published by Springer Nature that may be licensed from third parties.
If you would like to use or distribute our Springer Nature journal content to a wider audience or on a regular basis or in any other manner not expressly permitted by these Terms, please contact Springer Nature at

onlineservice@springernature.com