Hindi Word Sense Disambiguation using variants of Simplified Lesk measure

Satyendr Singh[†], Goldie Gabrani[†], Tanveer Siddiqui*

[†]School of Engineering & Technology, BML Munjal University, Gurgaon, India
[*]Department of Electronics & Communication, University of Allahabad, Allahabad, India

Abstract— **This paper evaluates the performance of a Lesk-like algorithm for Hindi Word Sense Disambiguation (WSD). The algorithm uses the similarity between the sense definition and the context of ambiguous word for disambiguation. Three different scoring functions have been investigated for measuring the similarity: direct overlap, frequency of matching words and frequency of matching words excluding the target word. We evaluate the effects of context window size, stop word elimination and stemming on Hindi WSD task. We also investigate the effect of number of senses on Hindi WSD task. The evaluation has been carried out on a manually created sense inventory consisting of 60 polysemous Hindi nouns. The maximum overall precision of 54.54% was observed for the case when both stemming and stop word removal was performed and frequency based scoring excluding the target word was used. The best case results in a significant improvement of 10.4% in precision and 21.3% in recall over the baseline performance. In general, we obtained decrease in precision with increasing number of senses.**

Index Terms— **Hindi word sense disambiguation, Dictionary based Word Sense Disambiguation, Lesk-based Word Sense Disambiguation.**

I. Introduction

Polysemy is a common property of all natural languages. Natural languages contain words bearing multiple meanings in different context. For example the Hindi noun 'हल' (hal) may mean solution or ploughing instrument, depending on the context. It is quite easy for human beings to arrive at the correct sense of word without even considering all the possible meanings. The words appearing in the context of an ambiguous word provide useful information about the correct sense of the ambiguous word. However, to determine the correct sense of an ambiguous word automatically is difficult. Word sense disambiguation (WSD) is the task of computationally choosing the correct sense of a polysemous word in a given context. It has been described as an "intermediate task" necessary for most natural language tasks [7] and has applications in machine translation, information retrieval and text categorization.

The WSD research mainly attempts to utilize the nearby context of an ambiguous word - the words appearing into the context or some of their features - to arrive at its correct sense. Much of these research focuses on English. The WSD research involving Hindi or other Indian languages is constrained by the lack of training and test dataset and benchmark. Due to the obvious differences between Hindi and English, the results obtained on English (or other European languages) cannot be generalized for Hindi (or Indian languages) without proper evaluation on Hindi data. In this paper, we attempt to evaluate a WSD algorithm for Hindi. The basic algorithm used in this work is similar to Lesk [10]. Following Vasilescu et al. [21], we call it simplified Lesk. We experiment with three different scoring functions to measure the similarity between the sense definition vector and the context vector and investigate the effects of the context window size, stemming and stop word removal on them. We further study the effect of number of senses on Hindi WSD task. The rest of the paper is organized as follows: In section II, related work is reviewed. In section III, WSD algorithm used in this work is discussed. The details of the data set used and the experiments conducted are provided in section IV. Results are discussed in section V. Finally, in section VI, we present our conclusion.

II. Related Work

WSD methods can be broadly categorized into dictionary-based and corpus-based (supervised and unsupervised) approaches. Dictionary-based approaches make use of information available in machine readable dictionaries, thesauri or lexical resources to disambiguate a word [1, 2, 4, 10, 21]. Corpus-based approaches use a corpus to extract information useful for disambiguation. Sense-tagged (supervised) [3, 9, 11] as well as raw corpora (unsupervised) [12, 22] have been used for disambiguation. The pioneer work in dictionary based approaches was done by Lesk [10] in which dictionary definitions were used to disambiguate a word. Each lexicon definition was represented as a bag of words occurring in the sense definition of target polyse-

[†]The author is with School of Engineering & Technology, BML Munjal University, Gurgaon, India
[*]Department of Electronics & Communication, University of Allahabad, Allahabad, India

mous word. Another bag of words was formed by extracting all the words occurring in the sense definitions of words appearing in the context of ambiguous word. The disambiguation was achieved using the overlap between the context bag and the sense bag. Since then several extensions to Lesk's algorithm have been proposed including [1, 2, 4, 21]. Banerjee and Pedersen [1] used glosses associated with synset and various semantic relations including hypernym, hyponym, holonym, meronym, troponym and attribute of each word in pair from English WordNet for disambiguation. Banerjee and Pedersen [2] explored new measure of semantic relatedness based on the number of overlaps in glosses. Their method extended the glosses of the concepts by including glosses of other concepts related using a concept hierarchy. Gaona et al. [4] used the word co-occurrences of the gloss and the context information for disambiguation. Vasilescu et al. [21] performed comparative evaluation of original Lesk algorithm, Lesk algorithm adapted to Wordnet and some variants of Lesk algorithm.

For Hindi language WSD reported work includes [8, 14, 15, 16, 17, 18, 19, 20]. Sinha et al. [20] employed a Lesk like algorithm for sense disambiguation. The sense bag was created by utilizing extended sense definitions comprising of synonyms, glosses, example sentences, and glosses and example sentences of hypernyms, hyponyms and meronyms of target polysemous noun. Context bag was created utilizing the nearby words of target polysemous noun. The overlap was computed between sense bag and context bag and the sense which maximized the score was assigned as winner sense. The evaluation was made on Hindi Corpora provided by Central Institute of Indian languages (CIIL) and accuracy values ranging from 40% to 70% was achieved. Khapra et al. [8] studied domain specific WSD, exploring dominant senses of words in specific domains for disambiguation. Nouns, adjectives and adverbs were extracted for English, Hindi and Marathi languages and they achieved an accuracy of 65% on F1-score for all the three languages. Singh et al. [19] explored Leacock Chodorow semantic relatedness measure for Hindi WSD. They obtained an overall average accuracy of 60.65% using this measure. Singh et al. [18] investigated Naïve Bayes classifier for Hindi WSD. They utilized rich features including local context, collocation, unordered list of words, nouns and vibhaktis. Singh and Siddiqui [17] studied the role of hypernym, hyponym, holonym and meronym relations in Hindi WSD. They obtained maximum improvement for single semantic relation using hyponym, resulting in overall improvement of 9.86% in precision. In [15], Singh and Siddiqui explored three WSD algorithms utilizing corpus statistics for Hindi WSD. The first algorithm utilized sense definitions and a sense tagged training corpus for performing disambiguation. The second algorithm utilized conditional probability of co-occurring words and phrases for disambiguation. The third algorithm was based on classification information model. Singh and Siddiqui [14] described the construction and details of Sense Annotated Hindi Corpus, a linguistic resource for Lexical Sample Hindi WSD task. In [16], Singh and Siddiqui attempted to capture the underlying similarity between the context and the sense definition in the presence of morphological variations. They observed 9.24% improvement in precision over the baseline on a small data set comprising 10 polysemous Hindi nouns. In this work, we have used frequency based scoring in addition to direct overlap. Moreover evaluation has been done on a sense annotated Hindi corpus consisting of 60 polysemous Hindi nouns.

III. WSD Algorithm

The WSD algorithm used in this work is simplified Lesk algorithm and is given in Fig. 1. The context vector is matched with extended sense definition. The extended sense definition consists of synsets, glosses and example sentences of polysemous word. The winner sense is the one having maximum overlap. The context of the target word is defined by the set of words appearing in a ± n window with the target word in the middle. The size of the context vector for a window size of n is 2n+1.

In order to study the effects of window size, test runs are conducted by varying the size from 5 to 25 in steps of 5. In order to study the effects of stemming we apply stemming on the sense definition and the context vector. Three different variants of scoring functions are used for measuring the similarity between the context vector and the sense definition vector. In the first, we use direct overlap, i.e., the number of matching words, for disambiguation. The second uses sum of the frequency of matching words. The third variant excludes the target word from the matching process. This was done in order to avoid disambiguating a word based on its own occurrence. In the rest of the paper we refer to these variants as Direct Overlap (DO), Frequency based Scoring (FS) and frequency-based scoring eXcluding target word (FX).

WSD algorithm
1. (a) Create context vector (C) consisting of all the words in a fixed window size n keeping the ambiguous word in the middle (Case 1).
or (b) Perform stemming on the sense definitions and test instances and then create context vector as in 1(a) (Case 2).
or (c) Remove stop words from sense definitions and from the test instances and then create context vector as discussed in 1(a) (Case 3).
or (d) Remove stop words and perform stemming on the sense definitions and test instances and then create context vector as in 1(a) (Case 4).
2. for i = 1 to number of senses do
    Create sense definition vector for sense i of the target word (S_i).
    $Score_i \leftarrow$ similarity (C, S_i)
3. return S_i for which score is maximum.

Computing score (Direct Overlap):
Similarity (C, S) // C is context vector & S is sense definition vector
sense_score ← 0
for each word w in C
  if w is in S
  sense_score = sense_score +1
return sense_score

Computing score (Frequency-based Scoring):
Similarity (C, S) // C is context vector & S is sense definition vector
sense_score ← 0
for each word w in C
  wcount ← frequency of w in S
  sense_score = sense_score + wcount
return sense_score

Fig.1 WSD Algorithm

IV. Data Set and Experiment

a. Data Set

One of the major hindrances for research in WSD of Hindi and Indian languages is lack of availability of standard sense tagged corpora for evaluation. Hence in order to evaluate the proposed algorithm, we created a Sense Annotated Hindi Corpus [13] containing 60 polysemous nouns (Table I). The Sense annotated Hindi corpus is available at Indian Language Technology Proliferation and Deployment Centre of Technology Development for Indian Languages (TDIL) portal. The sense inventory was derived from Hindi WordNet [6]. The contexts of ambiguous words were collected from Internet and Hindi Corpus [5] available at Centre for Indian Language Technology (CFILT), Indian Institute of Technology (IIT) Bombay. There are a total of 7506 test instances. The detail of the construction, translation, transliteration and statistics of the Sense Annotated Hindi Corpus is given in [14, 15]. The performance is measured in terms of precision and recall. Precision is defined as the ratio of the correctly disambiguated instances and total number of test instances answered for a word. Recall is defined as the ratio of the correctly disambiguated instances and the total number of test instances to be answered for a word.

Table 1. Sense Annotated Hindi Corpus

Number of Senses	Number of words	Word
2	36	अशोक (ashok), कांड (kaand), कोटा (kotaa), क्रिया (kriyaa), गल्ला (galla), गुना (guna), गुरु (guru), ग्राम (gram), घटना (ghatnaa), चंदा (chanda), चारा (chaaraa), जीना (jeena), जेठ (jeth), डब्बा (dabba), डाक (dak), ढाल (dhal), तान (taan), ताव (tao), तिल (til), तीर (teer), तुलसी (tulsi), दक्ष (daksh), दर (dar), दाद (daad), दाम (daam), धन (dhan), धुन (dhun), बाल (baal), माँग (maang), लाल (laal), विधि (vidhi), शेर (sher), सीमा (seema), सोना (sona), हल (hal), हार (haar)
3	19	अंग (ang), अंश (ansh), अचल (achal), उत्तर (uttar), कदम (kadam), कमान (kamaan), कुंभ (kumbh), क्वार्टर (quarter), खान (khan), चरण (charan), तेल (tel), थान (thaan), फल (phal), मत (mat), मात्रा (maatraa), वचन (vachan), वर्ग (varg), संक्रमण (sankraman), संबंध (sambandh)
4	3	कलम (kalam), धारा (dhaaraa), मूल (mool)
5	2	चाल (chaal), टीका (tika)

b. Experiments and Results

In order to evaluate the proposed algorithm and study the effects of stop word removal and stemming we perform four test runs for each variant of the algorithm. These test runs correspond to the following four cases:

(i) Without stop word removal and without Stemming (Case 1)
(ii) With stemming (Case 2)
(iii) With stop word removal (Case 3)
(iv) With stemming and stop word removal (Case 4)

Each test run is conducted by varying context window size from 5 to 25 in steps of 5. The baseline corresponds to direct overlap between the context and the sense definition (Case 1). Test run 2 performs stemming on both the context and the sense definitions (Case 2). In test run 3, we try to evaluate the effect of stop words in disambiguation. Hence, we create context vectors after dropping stop words (Case 3). Test run 4 uses both stemming and stop word removal (Case 4). For each variant of the algorithm, precision and recall values were computed for all the 60 nouns for each of the four cases on context window size of 5, 10, 15, 20 and 25. Context vectors of window size 10 for an instance of Hindi noun 'हल' (hal) as given in Fig. 2 for all four cases are given in Fig. 3.

उन्होंने कहा कि यदि इलाहाबाद हाईकोर्ट की लखनऊ पीठ के आदेश को ही मानें तो न्यायालय ने विवादित स्थल के एक तिहाई हिस्से को मुसलमानों को देने का आदेश दिया है। इस बीच मुस्लिम पक्ष के इस वयोवृद्ध पक्षकार से रविवार को बातचीत से मामले के हल के पैरोकार अवकाश प्राप्त न्यायाधीश पलक बसु ने भी मुलाकात की। बसु के अनुसार सुलह-समझौते से मसले के हल का कोई प्रयास छोड़ा नहीं जाना चाहिए।

{unhone kaha ki yadi allahabad highcourt ki lucknow peeth ke aadesh ko hi mane toh nayayalaya ne vivadit sathal ke ek tihai hisse ko musalmaano ko dene ka aadesh diya hai. Is beech muslim paksh ke is vayovradh pakshkaar se ravivaar ko baatcheet se maamle ke hal ke pairokaar avkaash prapt nayaayaadeesh palak basu ne bhi mulakaat ki. Basu ke anusaar sulah-samjhaute se masle ke hal ka koi prayas chora nahi jana chahiye}

Fig.2 Instance of 'हल' (hal)

Case 1: [इस, वयोवृद्ध, पक्षकार, से, रविवार, को, बातचीत, से, मामले, के, हल, के, पैरोकार, अवकाश, प्राप्त, न्यायाधीश, पलक, बसु, ने, भी, मुलाकात]

[is, vayovradh, pakshkaar, se, ravivaar, ko, baatcheet, se, maamle, ke, hal, ke, pairokaar, avkaash, prapt, nayaayaadeesh, palak, basu, ne, bhi, mulakaat]

Case 2: [इस, वयोवृद्ध, पक्षकार, स, रविवार, क, बातचीत, स, मामल, क, हल, क, पैरोकार, अवकाश, प्राप्त, न्यायाधीश, पलक, बस, न, भ, मुलाकात]

[is, vayovradh, pakshkaar, s, ravivaar, k, baatcheet, s, maaml, k, hal, k, pairokaar, avkaash, prapt, nayaayaadeesh, palak, bas, n, bh, mulakaat]

Case 3: [हिस्से, मुसलमानों, आदेश, मुस्लिम, पक्ष, वयोवृद्ध, पक्षकार, रविवार, बातचीत, मामले, हल, पैरोकार, अवकाश, प्राप्त, न्यायाधीश, पलक, बसु, मुलाकात, बसु, अनुसार, सुलह]

[hisse, musalmaano, aadesh, muslim, paksh, vayovradh, pakshkaar, ravivaar, baatcheet, maamle, hal, pairokaar, avkaash, prapt, nayaayaadeesh, palak, basu, mulakaat, basu, anusaar, sulah]

Case 4: [हिस्स, मुसलमान, आदेश, मुस्लिम, पक्ष, वयोवृद्ध, पक्षकार, रविवार, बातचीत, मामल, हल, पैरोकार, अवकाश, प्राप्त, न्यायाधीश, पलक, बस, मुलाकात, बस, अनुसार, सुलह]

[hiss, musalmaan, aadesh, muslim, paksh, vayovradh, pakshkaar, ravivaar, baatcheet, maaml, hal, pairokaar, avkaash, prapt, nayaayaadeesh, palak, bas, mulakaat, bas, anusaar, sulah]

Fig.3 Context vector for case 1, 2, 3 and 4 for window size 10

Table II and III shows the average precision and recall for 60 words averaged over context window size of 5, 10, 15, 20 and 25 for each case and variant. Table IV and V shows the average precision and recall for 60 words with respect to the context window size for each case and variant. In Sense Annotated Hindi Corpus, we have words having senses ranging from 2 to 5. There are 36 words having 2 senses, 19 words having 3 senses, 3 words having 4 senses and 2 words having 5 senses. Table VI and VII shows the average precision and recall of words with respect to the number of senses for each case and variant.

Table 2. Average Precision (Over 60 Words)

variant	Precision			
	Case 1	Case 2	Case 3	Case 4
DO	0.4743	0.4896	0.4787	0.5165
FS	0.4845	0.4825	0.4913	0.5195
FX	0.4940	0.4761	0.5064	0.5454

Table 3. Average Recall (Over 60 Words)

variant	Recall			
	Case 1	Case 2	Case 3	Case 4
DO	0.4288	0.4870	0.4336	0.5135
FS	0.4369	0.4802	0.4410	0.5167
FX	0.4469	0.4736	0.4578	0.5421

Table 4. Average Precision (Over 60 Words) With Respect To Context Window Size

		Precision				
	Variant	Context Window Size				
		5	10	15	20	25
Case 1	DO	0.4292	0.4610	0.4799	0.4978	0.5038
	FS	0.4618	0.4703	0.4865	0.4984	0.5053
	FX	0.4626	0.4839	0.4949	0.5099	0.5186
Case 2	DO	0.4364	0.4804	0.4955	0.5167	0.5191
	FS	0.4726	0.4843	0.4879	0.4842	0.4834
	FX	0.4626	0.4757	0.4822	0.4799	0.4804
Case 3	DO	0.3699	0.4490	0.4959	0.5258	0.5526
	FS	0.4321	0.4703	0.4996	0.5200	0.5342
	FX	0.3944	0.4707	0.5217	0.5583	0.5867
Case 4	DO	0.4137	0.4868	0.5291	0.5634	0.5897
	FS	0.4567	0.4995	0.5281	0.5510	0.5623
	FX	0.4467	.05075	0.5570	0.5957	0.6200

Table 5. Average Recall (Over 60 Words) With Respect To Context Window Size

		Recall				
	Var-iant	Context Window Size				
		5	10	15	20	25
Case 1	DO	0.3905	0.4190	0.4329	0.4476	0.4541
	FS	0.4179	0.4248	0.4383	0.4479	0.4555
	FX	0.4202	0.4391	0.4478	0.4595	0.4678
Case 2	DO	.04342	0.4779	0.4928	0.5140	0.5164
	FS	0.4706	0.4819	0.4856	0.4819	0.4810
	FX	0.4600	0.4732	0.4797	0.4774	0.4778
Case 3	DO	0.3394	0.4065	0.4473	0.4750	0.4996
	FS	0.3900	0.4224	0.4473	0.4663	0.4790
	FX	0.3595	0.4252	0.4704	0.5039	0.5302
Case 4	DO	0.4112	0.4838	0.5260	0.5602	0.5863
	FS	0.4542	0.4968	0.5253	0.5481	0.5593
	FX	0.4442	0.5044	0.5537	0.5922	0.6163

Table 6. Average Precision With Respect To Number Of Senses

		Precision			
		Number of Senses			
	variant	2	3	4	5
Case 1	DO	0.5337	0.4077	0.3154	0.2751
	FS	0.5368	0.4276	0.3356	0.3062
	FX	0.5470	0.4317	0.3842	0.2961
Case 2	DO	0.5428	0.4255	0.3928	0.2870
	FS	0.5471	0.4135	0.3145	0.2270
	FX	0.5369	0.4085	0.3107	0.2731
Case 3	DO	0.4860	0.4765	0.4514	0.4081
	FS	0.5548	0.4189	0.3059	0.3130
	FX	0.5182	0.4861	0.5566	0.4108
Case 4	DO	0.5332	0.4898	0.5440	0.4296
	FS	0.5773	0.4391	0.4406	0.3623
	FX	0.5676	0.5080	0.6424	0.3557

Table 7. Average Recall With Respect To Number Of Senses

		Recall			
		Number of Senses			
	variant	2	3	4	5
Case 1	DO	0.4772	0.3795	0.2960	0.2260
	FS	0.4797	0.3943	0.3243	0.2386
	FX	0.4914	0.3965	0.3689	0.2395
Case 2	DO	0.5401	0.4238	0.3840	0.2870
	FS	0.5445	0.4113	0.3142	0.2270
	FX	0.5341	0.4060	0.3096	0.2731
Case 3	DO	0.4345	0.4412	0.4216	0.3617
	FS	0.4910	0.3882	0.2924	0.2653
	FX	0.4630	0.4486	0.5186	0.3612
Case 4	DO	0.5303	0.4875	0.5320	0.4296
	FS	0.5743	0.4369	0.4341	0.3623
	FX	0.5645	0.5055	0.6302	0.3557

V. Discussion

As shown in Table II, case 4 outperforms for all the scoring functions. The maximum observed precision of 0.5454 corresponds to the case when both stemming and stop word elimination is applied (case 4) and frequency based scoring excluding the target word is used. The average precision (over all the words) for case 4 for direct overlap is 0.5165, which is 8.89% increase in precision over the baseline. The average precision (over all the words) for case 4, for frequency based scoring is 0.5195, which is 7.22% increase in precision over the baseline. The average precision (over

all the words) for case 4 for frequency based scoring excluding target word is 0.5454, which is 10.4% increase in precision over the baseline. The stop word elimination allows for more content words to contribute in disambiguation and stemming reduces morphological variants of these words to root forms thereby increasing the chances of overlap.

The results in Table IV suggest that increasing the size of context window in general improves the chances of correct disambiguation. This is because a larger window increases the number of content words some of which may be strong indicator of a particular sense. However, in some cases a drop in performance is observed. A word by word analysis suggests that this is due to the increased frequency of stop words in the context. As can be seen in Table IV, FX performs better in all the cases except case 2. In case 2 only stemming is applied which reduces morphological variations of words like 'के' (kay), 'की' (ki), 'का' (ka) and 'को' (ko) to same stem 'क' (k). A drop in accuracy is observed due to the increased match of karakas. In case 4 stop words are removed hence this problem does not occur.

The results in Table VI suggest that increasing the number of senses in general decreases the chances of correct disambiguation. We obtained maximum precision for words having 2 senses. We further obtained decrease in precision as the number of senses is increased with few exceptions. In case 4, we observed maximum average precision of 64.24% for words having 4 senses using frequency based scoring excluding target word. In case 4, both stemming and stop word removal have been applied. So, the context vector for this case constituted of words, which are strong indicator of a particular senses. The average precision observed in case 4 for words having 2 senses using frequency based scoring excluding target word is 56.76%, which is slightly lower as compared to average precision for words having 4 senses. One possible reason is in sense annotated Hindi Corpus there are only 3 words having 4 senses whereas the words having 2 senses are 36.

VI. Conclusion

In this paper, we evaluated the effect of context window size, stop word elimination and stemming on three variants of simplified Lesk algorithm for Hindi WSD. We further evaluated the effect of number of senses on Hindi WSD task. The maximum precision was observed for the case when both stemming and stop word elimination is performed. Stop word elimination results in more number of content words in context vector and stemming improves the chances of overlap by reducing the morphological variants of these words to the same stem. The content words thus become the dominant contributor to the score which otherwise is dominated by stop words. This results in improved accuracy. The increase in accuracy due to increase in context window size is also due to similar reasons. In general, the disambiguation accuracy decreases with the increasing number of senses.

References

1. Banerjee, S. and Pederson, T.: An Adapted Lesk Algorithm for Word Sense Disambiguation using WordNet. In Proceedings of the Third International Conference on Computational Linguistics and Intelligent Text Processing, pp. 136-145 (2002)
2. Banerjee, S. and Pederson, T.: Extended gloss overlaps as a measure of semantic relatedness. In Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence, Acapulco, Mexico, pp. 805-810 (2003)
3. Gale, W. A., Church, K. and Yarowsky, D.: A method for disambiguation word senses in a large corpus, In Journal of Computer and the Humanities, vol. 26, pp. 415-439 (1992)
4. Gaona, M. A. R., Gelbukh, A. and Bandyopadhyay, S.: Web-based variant of the Lesk approach to Word Sense Disambiguation. In Mexican International Conference on Artificial Intelligence, pp. 103-107 (2009)
5. Hindi Corpus http://www.cfilt.iitb.ac.in/Downloads.html
6. Hindi WordNet http://www.cfilt.iitb.ac.in/wordnet/webhwn/wn.php
7. Ide, N. and Veronis, J.: Word Sense Disambiguation: The State of the Art. Computational Linguistics, vol. 24, issue 1, pp. 1-40 (1998)
8. Khapra, M. M., Bhattacharyya, P., Chauhan, S., Nair, S. and Sharma A.: Domain Specific Iterative Word Sense Disambiguation in a Multilingual Setting. In Proceedings of International Conference on NLP (ICON 08), Pune India (2008)
9. Lee, Y. K., Ng, H. T. and Chia, T. K.: Supervised Word Sense Disambiguation with Support Vector Machines and Multiple Knowledge Sources. In SENSEVAL-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text, Barcelona, Spain, pp.137-140 (2004)
10. Lesk, M.: Automatic Sense Disambiguation using Machine Readable Dictionaries: How to Tell a Pine Cone from an Ice Cream Cone. In Proceedings of the 5th annual international conference on Systems documentation SIGDOC Toronto, Ontario, pp.24-26 (1986)
11. Ng, H. T. and Lee, H. B.: Integrating multiple knowledge sources to disambiguate word sense: An exemplar based approach. In Proceedings of the 34th Annual meeting for the Association for Computational Linguistics, pp. 40-47 (1996)
12. Resnik, P.: Selectional preference and sense disambiguation. In Proceedings of the ACL SIGLEX workshop on tagging text with lexical semantics: Why, What and How?, pp. 52-57 (1997)
13. Sense Annotated Hindi Corpus http://www.tdil-dc.in/index.php?option=com_download&task=showresourceDetails&toolid=1472&lang=en
14. Singh, S. and Siddiqui, T. J.: Sense Annotated Hindi Corpus. In the 20th International Conference on Asian Language Processing, Tainan, Taiwan, pp. 22-25 (2016)
15. Singh, S. and Siddiqui, T. J.: Utilizing Corpus Statistics for Hindi Word Sense Disambiguation. In International Arab Journal of Information Technology, volume 12, no. 6A, pp. 755 – 763 (2015)
16. Singh, S. and Siddiqui, T. J.: Evaluating Effect of Context Window Size, Stemming and Stop Word Removal on Hindi Word Sense Disambiguation. In Proceedings of the International Conference on Information Retrieval and Knowledge Management, Malaysia, pp. 1-5 (2012)
17. Singh, S. and Siddiqui, T. J.: Role of Semantic Relations in Hindi Word Sense Disambiguation. In Proceedings of International Conference on Information and Communication

Technologies (ICICT 2014), Kochi, India, Procedia Computer Science, vol. 16, pp. 240-248 (2015)
18. Singh, S., Siddiqui, T. J. and Sharma, S. K.: Naïve Bayes classifier for Hindi Word Sense Disambiguation. In Proceedings of 7th ACM India Compute Conference, Article No. 1, pp. 1-9 (2014)
19. Singh, S., Singh, V. K. and Siddiqui, T. J.: Hindi Word Sense Disambiguation using Semantic Relatedness measure. In Proceedings of 7th Multi-Disciplinary workshop on Artificial Intelligence, Krabi, Thailand, pp. 247-256 (2013)
20. Sinha, M., Kumar, M., Pande, P., Kashyap, L. and Bhattacharyya, P.: Hindi Word Sense Disambiguation. In International Symposium on Machine Translation, Natural Language Processing and Translation Support Systems, Delhi, India (2004)
21. Vasilescu, F., Langlasi, P. and Lapalme, G.: Evaluating Variants of the Lesk Approach for Disambiguating Words. In Proceedings of the Language Resources and Evaluation, pp. 633 - 636 (2004)
22. Yarowsky, D.: Unsupervised word sense disambiguation rivaling supervised methods. In Proceedings of the 33rd Annual meeting of the Association for Computational Linguistics, pp. 189-196 (1995)

Satyendr Singh is currently Assistant Professor at BML Munjal University, Gurgaon, India. He received B.E. in Computer Science and Engineering from Ch. Charan Singh University, Meerut, India in 2000. He obtained M.E. in Computer Science and Engineering from Panjab University, Chandigarh, India in 2008. He obtained Ph.D. in Computer Science from University of Allahabad, Allahabad, India in 2015. His research interests include natural language processing and machine learning.

Goldie Gabrani is currently Professor at BML Munjal University, Gurgaon, India. She received B.E., M.E. and Ph.D. in Computer Engineering from University of Delhi, Delhi, India. She has teaching, research and industry experience of more than 30 years. Her research interest includes Artificial Intelligence, Distributed Computing and Data Analytics.

Tanveer Siddiqui is currently Associate Professor at University of Allahabad, Allahabad, India. She obtained M.Sc. and Ph.D degree in Computer Science from University of Allahabad, Allahabad, India. She has experience of teaching and research of more than 16 years in the area of Computer Science and Information Technology. Her research interests includes natural language processing, human computer interaction and information retrieval.