Kannada word sense disambiguation using semantic relations

B.H.Manjunatha Kumar1 M.Siddappa2

1Assistant Professor
Department of Computer Science and Engineering
Sri Siddhartha Institute of Technology, Tumakuru, Karnataka, India

2Dean (Academics)
Sri Siddhartha Academy of Higher Education, Tumakuru, Karnataka, India

bhm.nlp@gmail.com

Abstract. In this paper, a method is proposed to use the semantic relations of words to solve Word Sense Disambiguation (WSD) for the Kannada language. Semantic relationships like hypernymy and hyponymy are used in finding the overlaps between two concepts. Integrating the idea proposed in this paper with original Lesk’s technique improves the accuracy of the Lesk’s algorithm. Indo-WordNet is used as the lexical resource to gain the knowledge about the semantic relations of words and this knowledge is exploited to disambiguate the target word of the Kannada language. Since the approach used in this paper is unsupervised, so it can be scalable to other languages also.

1. Introduction
Word Sense Disambiguation is the process of determining the desired sense of the target word according to the context where it is used. WSD is viewed as an essential and also an useful process in most of the language technology applications like machine translation [1], information extraction or retrieval, knowledge acquisition or meaning [2], lexicography [3], and semantic interpretation, and is now even more important in latest research areas like bioinformatics [4] and Semantic Web. In our previous work, three approaches were proposed for disambiguating the Kannada word based on an adaptation of dictionary based Lesk’s WSD technique [6]. In all the three approaches only the meaning of the words are considered for disambiguating the ambiguous word. Accuracy of this technique can be increased by exploiting the knowledge about the semantic relations of words along with their meanings. This paper proceeds with an overview about Indo-WordNet and a brief explanation of Lesk’s algorithm. This is accompanied by a precise presentation of proposed algorithm for the Kannada language and a discussion on its outcomes.

2. Related work

2.1. Indo-Wordnet
WordNet is similar to a dictionary containing words and their meanings. However, it is different from the traditional dictionary in several ways. For example, in WordNet words are organized semantically rather than alphabetically. Synonymous words are collected together to build synsets or synonym sets. Each synset of this kind represents a distinct concept or sense. IndoWordNet is a linked wordnet of
major Indian languages, viz, Assamese, Bangla, Bodo, Gujarati, Hindi, Kannada, Kashmiri, Konkani, Malayalam, Manipuri, Marathi, Nepali, Oriya, Punjabi, Sanskrit, Tamil, Telugu and Urdu [5]. These WordNets are created by making use of expansion strategy from WordNets of Hindi and English language. Every single entry in the IndoWordNet contains the components like Synonymy, Gloss, and example sentence along with an associated lexicon in different Indian languages an also from the English language. The Indowordnet project is a composite of North East WordNet, Dravidian WordNet, and Indradhanush project, which are funded by technology development of Indian language programme.

Words having multiple senses are grouped as either homonymous or polysemous. A word is said to be homonymous when its meanings sense completely different things having the same spelling. If the senses of the word are shades of the same basic meaning, then they are termed as polysemous words. WordNet does not differentiate between polysemous and homonymous words. Words having only one sense are called as monosemous.

Wordnet contains information regarding the words that fall under four parts – of – speech (POS): nouns, verbs, adjectives, and adverbs. Currently, IndoWordNet has nouns, verbs, adjectives, and adverbs. We have performed our experiments by using IndoWordNet as the lexical database so we can disambiguate only the words which appear in certain synset in IndoWordNet. In IndoWordNet, each synset is connected with the following elements:

- **Synset ID:** Unique ID assigned to every synset.
- **Gloss or definition:** The gloss is made up of a brief explanation of the meaning of the concept defined by the synset.
- **Example sentences:** These sentences show how the words in the synset can be used in framing the natural language sentences.
- **Hypernyms (hype):** List of words whose meaning includes a group of other words. For example, color is a hypernym of red.
- **Hyponyms (hypo):** List of words of more specific meaning than a general term applicable to it. For example, a spoon is a hyponym of cutlery.
- **Antonyms:** List of words opposite in meaning to the given word.
- **Onto tree:** Gives the details about the ontology of the given word.

WordNet describes various lexical and semantic relations between synsets and words. Lexical relations define a relation between two words within the synsets of WordNet. Whereas the relationship between the two synsets is defined by semantic relations. Semantic relations relate words in one synset to all the words in another synset. In IndoWordnet, the majority of the relations usually do not go over the parts of speech (POS) boundaries, so relations are defined among the words or synsets that fall under same POS.

2.2. **Lesk Algorithm**

Lesk algorithm is the first semantic disambiguation algorithm developed in 1986 using traditional dictionaries for WSD. The main concept behind this algorithm is, most likely sense of the word with respect to the given context is determined by measuring contextual overlap among machine-readable dictionary definitions or glosses of different senses of the target or an ambiguous word [7]. The sense whose definition has the highest number of common words with the definitions of other words is assigned as the correct sense to the ambiguous word. The important limitation of the original Lesk method is that dictionary definitions are small in length and may not contain sufficient vocabulary. To overcome from the limitation of original Lesk algorithm, three approaches were proposed based on the original Lesk’s idea in [6]. Indo-WordNet was used as the main lexical resource to extend the length of the gloss. The sense definition of the ambiguous word used in these approaches does not contain all the possible words which can be used with the given target word. If the gloss is not strengthened enough then the accuracy of the system also decreases. These observations made us to extend the gloss of the sense by exploiting the semantic hierarchy of the Indo-WordNet.
3. Changes made to the original Lesk’s technique

Lesk algorithm was developed before the development of WordNet. Traditional dictionaries were used as the main lexical resource. Satanjeev Banerjee and Ted Pedersen [8] modified the Lesk’s technique to design a WSD system using English WordNet. They used glosses of related words along with the glosses of the ambiguous word itself to decide the most probable sense of an ambiguous word whereas original Lesk technique takes only the gloss of a word sense. In this work different semantic relations like hypernymy, hyponymy and antonymy are used to find the related words of the words present in the context window. Here we have considered the context window of size 3 words and IndoWordNet is used as the lexical database so only the words which appear in the synsets of IndoWordNet can be disambiguated.

Two strategies are followed for selecting the gloss pairs while finding the overlaps between the glosses of two words. They are heterogeneous and homogeneous. In the heterogeneous method, each gloss related to the sense considered for the first word is compared with each gloss related to the sense considered for the second word. In the homogeneous method, we perform the comparison between glosses of the two words as well as between the glosses of the synsets which are associated with these senses using same semantic relations. Heterogeneous and homogeneous strategies are explained in the following example:

Consider two Kannada words ಹೊಸುತ್ತಕು and ಹೊಸುನು. Word ಹೊಸುತ್ತಕು has only one sense whose synset id is 13720 and belongs to the POS noun. Sense of the word ಹೊಸುತ್ತಕು is denoted as ಹೊಸುತ್ತಕು / N / 13720. Word ಹೊಸುನು has 6 senses whose synset ids are 1144, 8463, 12490, 12491, 14271 and 27697. For explanation purpose we consider the synset id 12490 of the word ಹೊಸುನು. Its corresponding sense is denoted as ಹೊಸುನು / N / 12490. gloss(ಹೊಸುತ್ತಕು / N / 12490), hype(ಹೊಸುತ್ತಕು / N / 12490), hypo(ಹೊಸುತ್ತಕು / N / 12490), gloss(hype(ಹೊಸುನು / N / 12490)) and gloss(hypo(ಹೊಸುನು / N / 12490)) are as defined earlier.

For the word ಹೊಸುನು:

\[\text{gloss (ಹೊಸುನು / N / 12490) = ಹೊಸುನು, ಹೊಸುನು, ಹೊಸುನು, ಹೊಸುನು, ಹೊಸುನು, ಹೊಸುನು} \]
\[\text{hype (ಹೊಸುನು / N / 13720) = \{ ಹೊಸುನು, ಹೊಸುನು \}} \]
\[\text{hypo (ಹೊಸುನು / N / 13720) = \{ \}} \]
\[\text{gloss (hype(ಹೊಸುನು / N / 12490)) = ಹೊಸುನು, ಹೊಸುನು, ಹೊಸುನು, ಹೊಸುನು, ಹೊಸುನು, ಹೊಸುನು} \]
\[\text{gloss (hypo(ಹೊಸುನು / N / 13720)) is an empty string.} \]

Figure 1 and Figure 2 show the heterogeneous and homogeneous schemes diagrammatically. In these figures ovals represent the synsets, synset members are enclosed in \{\} and the gloss is written within double quotes. Double headed arrow line denotes the different gloss-gloss comparisons. We perform 4 comparisons in Figure 1 by comparing both the glosses related to ಹೊಸುತ್ತಕು / N / 13720 with each glosses related to ಹೊಸುನು / N / 12490. gloss(ಹೊಸುನು/N/12490) and gloss (ಹೊಸುತ್ತಕು / N / 13720) comparison gives an overlap of a word ಹೊಸುನು which is same as the output of the original Lesk technique. But other three comparisons of the glosses created using various semantic relations give a total of 10 common words. There is a five word overlap in the comparison between hype(ಹೊಸುನು/N/12490) and hype (ಹೊಸುತ್ತಕು / N / 13720) glosses. It confirms that the two senses ಹೊಸುನು / N / 13720 and ಹೊಸುನು/N/12490 are strongly related.

As depicted in Figure 2 two comparisons performed in the homogeneous method and produce 8 common words, which is less than the heterogeneous method. However, homogeneous scheme performs less number of comparisons. In general, if there exists a first-word sense with n glosses and second-word sense with m glosses, then the heterogeneous method performs n × m comparisons and
The homogeneous method performs min (n, m) comparisons. The homogeneous method can be viewed as an immediately extended version of the original Lesk technique. In the original Lesk technique, only the gloss of the target word was compared with the glosses of the words present in the context window. In a similar way, in a homogeneous method for comparison, only the glosses of the words which are related to each other using the same semantic relation are considered. Whereas in the heterogeneous method, all possible glosses and semantic relations exists between a pair of words are considered for comparison.

Figure 1. Homogeneous scheme of gloss pair selection.

Figure 2. Homogeneous scheme of gloss pair selection.
4. WSD Algorithm using Semantic Relations

In the proposed approach a context window of size three words containing one target word and other two its surrounding words is used. For each word in the context window, the glosses are defined using the different semantic relations. Either heterogeneous method or homogeneous method can be used for selecting the gloss pairs. Senses are taken from Indo-WrdNet. Each word of the context window may have one or more than one synsets. These synsets are grouped into clusters based on the ontology of the corresponding word. If a word having more than one synset, then we compare the ontology of all the synsets and those having similar ontology are grouped into a cluster. Each synset present in a cluster has the same hypernyms, hyponyms, and holonyms. Hence the number of synsets need to consider is equal to the number of clusters. This observation made on Indo-WordNet has helped us to reduce the complexity during the implementation of the algorithms. All the three approaches are explained in the following sections.

The proposed strategy makes an attempt to find the best combination of senses for the words present in the context window. This best combination will disambiguate the target word and simultaneously disambiguate other words present in the context window. Here sense assigned to the target depends on the sense assigned to its surrounding words and vice-versa. Let N is the size of the context window and W_i, 1 ≤ i ≤ N is the Indo-WordNet word present in the context window. Each word W_i may have one or more than one senses. We consider all the possible senses of the words present in the context window. |W_i| is the total number of senses of the word W_i and \( \prod_{i=1}^{N} |W_i| \) are the total number of possible combinations, each combination is called as the candidate combination.

Algorithm: WSD algorithm using semantic relations

Aim: Disambiguating the ambiguous word of the Kannada language.
Input: Kannada sentence containing a single target word.
Output: Correct Indo-WordNet sense for the target word.

Define a context window of size N
best_score = 0
for i \( \leftarrow 1 \) to \( \prod_{i=1}^{N} |W_i| \)
    w[1 ... N] = next_candidate-combination(w)
    candidate_score = 0
    for j \( \leftarrow 1 \) to N-1
        for k \( \leftarrow i+1 \) to N
            for each relation R_1 in the set of semantic relations
                for each relation R_2 in the set of semantic relations
                    candidate_score = candidate_score + get_score(gloss(R_1(w[i])), gloss(R_2(w[j])))
                end for
            end for
        end for
    end for
    if candidate_score > best_score
        best_score = candidate_score
        best_candidate_combination[1 ... N] = w[1 ... N]
    end if
end for
print best_candidate_combination and best_score

In the proposed algorithm, for each candidate combination, a candidate score is calculated by comparing different glosses related to them. Select the gloss-pairs for comparison by following either the heterogeneous method or the homogeneous method. Square and add the number of common words present in each comparison. The scores calculated for every combination of senses are added together.
to get the candidate score of the combination. The combination which attains the maximum score is selected as the winning combination and every word belongs to the context window will be assigned with their related sense according to the winning combination.

5. Experimental Results and Evaluation

The proposed approach is implemented using Indo-WordNet as the primary lexical resource. For experiments, we used the dataset given in Table 1 and input sentences are taken from the Kannada blogs, e-newspapers, Kannada articles etc. Implementation of the proposed algorithm is illustrated with the following example.

Consider the sentence ಬೆಳೆಯು ವರ್ಷ ಯಾನ್ಯ ಸಣಳುವಿನ, ಪುತ್ತುಭಾನು ಹಿಂಜಿಗೃಹ್ಣ and a context window of size three words. ಬೆಳೆಯು, ವರ್ಷ, and ಹಿಂಜಿಗೃಹ್ಣ are the three words of the context window.

Target word in this example is ಹಿಂಜಿಗೃಹ್ಣ. The senses and gloss of the context window words is shown in the Table 2.

Let A = ಬೆಳೆಯು/N/4500, B = ವರ್ಷ/N/27538, C = ಹಿಂಜಿಗೃಹ್ಣ/N/7954, D = ವರ್ಷ/N/(3869) and E = ಹಿಂಜಿಗೃಹ್ಣ/N/7030. In this example we can form a total of 4 distinct combinations:

Candidate_combination 1: A – B – D
Candidate_combination 2: A – C – D
Candidate_combination 3: A – B – E
Candidate_combination 4: A – C – E

We defined the hypernymy and hyponymy relations for the words of the context window. Hypernymy relation exists for all the three words and hyponymy relation exists for all the words except ವರ್ಷ/N/27538 and ಹಿಂಜಿಗೃಹ್ಣ/N/7030 in Indo_WordNet. We calculated the combination score of all the candidate combinations, the combination A – B – D got the highest score. This combination decides the right sense of the target word ಹಿಂಜಿಗೃಹ್ಣ as ವರ್ಷ/N/27538 means rain and correct sense to the non-target word ಹಿಂಜಿಗೃಹ್ಣ as ವರ್ಷ/N/(3869+6809) means flood or rain fall in the given context. The scores and the overlaps calculated by our system for the candidate_combination 1 are shown in the Table 3.

The proposed algorithm is implemented and evaluated on the data set shown in the Table 1. For all the experiments context window of size 3 words is used. The proposed algorithm has returned the answer for all the target words hence we got the same score for all the three performance evaluation metrics. The precision, F-measure, and recall values are given in Table 4.

| Table 1. Dataset |
|------------------|
| **Words** | **Transliteration** | **Translation** | **Number of Senses** |
| ಬೆಳೆ | Bele | Corp or growth | 2 |
| ವರ್ಷ | Varsha | Year or rain | 2 |
| ಯಾನ್ಯ | Yantra | Machine or ju-ju | 2 |
| ಗುರು | Guru | Teacher or Name of the planet | 2 |
| ಪುತ್ತುಭಾನು | Uttara | Answer or North direction | 2 |
| ಹಿಂಜಿಗೃಹ್ಣ | Charakha | Spinning wheel or Name of the scientist | 2 |
| ಸಣಳುವಿನ | There | Open or screen | 3 |
| ಹಿಂಜಿಗೃಹ್ಣ | Kaditha | itching or bite | 3 |
| Word | Synset ID | POS  | Gloss                                           |
|------|-----------|------|-------------------------------------------------|
| ಮೂಲ | 4500      | Noun | ಭಾವಿಸಿದ್ದು ವಿಭಾಗ ಮತ್ತು ಕರ್ಮದ ರೇಖೆಗಳು ಮೂಲಕ ಸಂಬಂಧಿಸಿದ್ದು ಮಾತ್ರ. |
| ವರಗಳು | 27538    | Noun | ವಸ್ತುಗಳ ಕೇಂದ್ರ ಮತ್ತು ಭಾವಿಸಿದ್ದುವಿಕೆಗಳು. |
| ಬಾಬುರು | 7954      | Noun | ಬಾಬುರುಗಳ ಸಹಾಯಕರು ಹಾಗು ಯುದ್ಧದ ಪಾಲಕರಾಗಿದ್ದು. |

Table 2. Synset details of context window words
Table 3. Scores and overlap computation for the candidate_combination 1

| First gloss | Second gloss | Overlap string | Score |
|-------------|--------------|----------------|-------|
| A           | B            |                | 0     |
| hype(A)     | B            |                | 0     |
| hypo(A)     | B            |                | 0     |
| A           | D            | ಇದ್ದೆ          | 1     |
| hype(A)     | D            |                | 0     |
| hypo(A)     | D            |                | 0     |
| B           | D            | ಬೆಳ್ಳಿ, ಬೆಳ್ಳಿ | 2     |
| hype(B)     | D            | ಬೆಳ್ಳಿ, ಬೆಳ್ಳಿ, ಮಹಾದಹಬಲ | 3     |
| hype(A)     | hype(B)      |                | 0     |
| hypo(A)     | hype(B)      |                | 0     |
| hype(A)     | hype(D)      |                | 0     |
| hype(A)     | hypo(D)      |                | 0     |
| hypo(A)     | hype(D)      |                | 0     |
| hypo(A)     | hypo(D)      |                | 0     |
| hype(B)     | hypo(D)      |                | 0     |
| hype(B)     | hypo(D)      | ಬೆಳ್ಳಿ, ಬೆಳ್ಳಿ | 2     |

Total score for the candidate_combination 1: A – B – D 8

Table 4. Evaluation of WSD algorithm with non-target words comparison

| Performance evaluation metrics | Value |
|--------------------------------|-------|
| Precision                      | 0.73  |
| Recall                         | 0.73  |
| F-measure                      | 0.73  |

6. Observations made during the implementation

Following observations have been made during the implementation of the algorithm:
- While implementing this algorithm, sometimes we get a combination score of 0 (no common words found). It indicates that the system is not able to provide any proof for any sense of the ambiguous word. In this situation, we avoided reporting any answer.
- Heterogeneous method of gloss-pair selection for comparison increases the time complexity of the algorithm. If there exist S senses per word and a context window of size N, then the total number of sets of synsets considered for comparison is
  \[ S^2 \times \frac{N \times (N - 1)}{2} \] (1)
Value of the Eq. (1) is exponential in nature. As the N increases, the time complexity of the algorithm also increases exponentially.

- Stemming is performed before comparing the words present in the glosses using unsupervised Kannada stemmer [9, 10]. This process will convert the inflected word into a root word and it avoids missing the matches when two words of the same root word are compared. For example, in the following two Kannada sentences: 

\[
\text{ನೂಡಿನ ಮರಾಟ ಅಧ್ಯಪಣ ಶಿಕ್ಷಣೆ ಮತ್ತು}
\]

\[
	ext{ನೂಡಿನ ಮರಾಟ ಅಧ್ಯಪಣ ಶಿಕ್ಷಣೆ ಮತ್ತು}
\]

If the strings of these two sentences are not stemmed, then we will get only one root word as the common word even though there exists words and are two forms of the same root word and we will miss the match of size two words.

7. Comparison with other WSD algorithms
The results of the proposed algorithm are compared with the results of the pure Lesk technique, and noticed that a great increase in the accuracy by the method proposed in this paper. The comparison is shown in Table 5. In WSD algorithm using semantic relations an error rate is reduced by 16% compared to the results of pure Lesk’s algorithm. This error rate reduction has increased the accuracy of the system from 0.53 to 0.73.

| Algorithm | Precision (P) | Recall (R) | F-Score |
|-----------|--------------|------------|---------|
| Lesk evaluation by Banerjee S [8] | 0.183 | 0.183 | 0.183 |
| Kannada WSD Finding the Overlaps between the concepts [6] | 0.57 | 0.53 | 0.55 |
| WSD algorithm using semantic relations | 0.73 | 0.73 | 0.73 |

8. Conclusion and future work
The WSD algorithm proposed in this paper uses semantic relations to determine the right sense for the target word. First, we made an attempt to strengthen the gloss considered for the comparison by adding the gloss of the related words of the target word. We then presented the heterogeneous and homogeneous strategies for selecting the gloss pairs for comparison. We then designed a WSD algorithm which makes the additional comparison between non-target words. The common words found in this comparison give the great evidence to assign the sense which is used for the comparison to the target word. If the input sentences are long and few words of the sentences are strongly related to the target word, then it is necessary to find those related words for the algorithm to work well. At present we do not have any automatic way to decide which word of the sentences is strongly related to the target word.

9. References
[1] Carpuat M and Wu D 2005 Word sense disambiguation vs. statistical machine translation Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics - ACL '05 doi:10.3115/1219840.1219888
[2] Resnik P 2006 Word sense disambiguation in natural language processing applications In Agirre E., & Edmonds, P. (Eds.), Word Sense Disambiguation chap.11 pages 299–337
[3] Kilgarriff A The hard parts of lexicography 1998 International Journal of Lexicography 11(1) 51-54 doi:10.1093/ijl/11.1.51
[4] Pesaranaghad A, Pesaranaghad A and Mustapha N Word Sense Disambiguation for Biomedical Text Mining Using Definition-Based Semantic Relatedness and Similarity Measures 2014 International Journal of Bioscience, Biochemistry and Bioinformatics 4(4) 280-283 doi:10.7763/ijbb.2014.v4.356
[5] Bhattacharyya P 2016 IndoWordNet The WordNet in Indian Languages 1-18 doi:10.1007/978-981-10-1909-8_1
[6] Manjunatha Kumar B H, Siddappa M and Prakash J Kannada Word Sense Disambiguation by Finding the Overlaps between the Concepts 2018 *International Journal of Engineering & Technology* 7 189-192 doi:10.14419/ijet.v7i2.6.10565

[7] Lesk and Michael 1986 Automatic Sense Disambiguation Using Machine Readable Dictionaries *Proceedings of the 5th Annual International Conference on Systems Documentation - SIGDOC 86* doi:10.1145/318723.318728

[8] Banerjee S and Pedersen T 2002 An Adapted Lesk Algorithm for Word Sense Disambiguation Using WordNet *Computational Linguistics and Intelligent Text Processing Lecture Notes in Computer Science* 136-145 doi:10.1007/3-540-45715-1_11

[9] Manjunatha Kumar B H, Siddappa M and Prakash J Unsupervised Kannada Stemmer using Partial Lemmatizer and Indo – Word Net 2017 *International Journal of Engineering Research and Applications (IJERA)* 7

[10] Manjunatha Kumar B H and Mounika S 2016 Unsupervised Stemmer for Kannada Language *International Journal of Innovative Research in Technology (IJIRT)* 3