Adapted Lesk Algorithm based Word Sense Disambiguation using the Context Information

Manish Kumar[1]
Department of Information Technology, SBPDCL
Bihar, India

Prasenjit Mukherjee[2]
Manik Hendre[3], Manish Godse[4]
Dept. of Analytics and IT
Pune Institute of Business management, Pune, India

Baisakhi Chakraborty[5]
Dept. of Computer Science and Engineering, NIT
Durgapur, India

***Abstract*—The process of identifying the meaning of a polysemous word correctly from a given context is known as the Word Sense Disambiguation (WSD) in natural language processing (NLP). Adapted Lesk algorithm based system is proposed which makes use of knowledge based approach. This work utilizes WordNet as the knowledge source (lexical database). The proposed system has three units – Input query, Pre-Processing and WSD classifier. Task of input query is to take the inputs sentence (which is an unstructured query) from the user and render it to the pre-processing unit. Pre-processing unit will convert the received unstructured query into a structured query by adding some features such as Part of Speech (POS) tagging, grammatical identification (Subject, Verb, and Object) and this structured query is transferred to the WSD classifier. WSD classifier uniquely identifies the sense of the polysemous word using the context information of the query and the lexical database.**

Keywords*—*Word Sense Disambiguation; natural language processing; WordNet; context; machine translation

I. Introduction

Natural language processing (NLP) is the field of computational linguistics or artificial intelligence that is concerned with the interaction between computers and human (natural) languages [1]. Natural language processing plays the important role in communication between human and machine. Word sense disambiguation is an important area of natural language processing and its application is related to find out the correct sense of an ambiguous word that is being used in a sentence. Many supervised and unsupervised algorithms have been developed on word sense detection as in [2]. In this field, a computer system is programmed in such a way that it is able to process a query provided in natural language and determine its correct semantics (meaning). Query is provided in the form of a sentence or a paragraph or text document. The semantics of a sentence depends on the semantics of its constituent words which are the smallest units of a sentence. Most of the words used in natural languages are associated with multiple meanings and these meanings vary frequently with the change in the contexts. Word with more than one meanings or senses are called polysemous words in the field of natural language processing and creates the problem of sense ambiguity. This work proposes a system such that it can correctly identify the meaning of the word (s) for the given context (sentence). Often a polysemous word has different meanings in different contexts. For example, the English word “bank” is associated with multiple meanings: “A financial institution”, “slopping land besides the water body”, or “have faith or confidence in” and many more as referred in the Princeton WordNet 3.0 [3]. The process of identifying the meaning of a polysemous word correctly in a given context is known as Word Sense Disambiguation (WSD).

Example 1

C1: I went to the **bank** to withdraw some money.

C2: Kolkata is situated at the **bank** of river Hugli.

Example 2

C3: **Cricket** is type of game.

C4: **Cricket** is a type of Insect.

Clearly, the word “bank” has two different meanings in the contexts C1 and C2 which are “a financial bank” and “land besides the river or sea” respectively. Similarly the word Cricket has also two different meanings in the contexts C3 and C4 respectively.

In the work proposed, knowledge based approach has been used for sense disambiguation. Natural language applications are using word sense disambiguation that is essential part in semantic analysis. Many models have been developed in word sense disambiguation where word space model is an effective model. This model represent the context vectors and sense vector in word vector space. Vector space is an important component to sense of a word as in [4]. Disambiguation of word sense using knowledge based approach can be done in two ways: Overlap method and Graph method. One of the pioneer works in overlap method is Lesk algorithm [5] that counts common words between two glosses (word definitions) to identify correct sense in the context. Gloss plays the vital role in the Lesk algorithm, which expresses two types of information: information about set of entries of all possible meanings and contextual information of target word. For the given pair of words, Lesk algorithm extracts meanings from lexical database and selects that sense as final sense which has the maximum overlap/common words/ co-occurred words. Lesk algorithm adapted Oxford Advanced Learner’s Dictionary as lexical database (also called as sense inventory). After some years later, two variants of Lesk [6] [7] have been proposed – “*Simplified*” version of Lesk algorithm [8] and Adapted Lesk algorithm [6]. The adapted Lesk algorithm adapted WordNet as lexical database and used the semantic relationships defined in the WordNet such as Hypernym, Hyponyms, Troponym, Meronyms, etc. Both Algorithms outperform the Lesk one as proved by Vosilescue et al. [9].

The specific meaning determination of an ambiguous word according to the context is a main task of word sense disambiguation. Mohammad Shibli Kaysar et al. [10] have introduced a system that is based on Bengali word sense disambiguation. A FP-Growth algorithm and Apriori algorithm have been proposed by the Authors on Bengali word sense disambiguation. The proposed system has been tested and 80% good result has been generated from ambiguous words as in [10]. Word sense disambiguation in Hindi language is limited. Anidhya Athaiya et al. [11] have approached Hindi language based word sense disambiguation that is genetic algorithm based. The proposed window is dynamic and feature of this window is containing vague word with left and right expression. The possible senses of an ambiguous Hindi word can be extracted from Hindi WordNet that is created by the IIT, Bombay as in [11].

The proposed work is an extension of adapted Lesk algorithm. Glosses provide the key information since they express the meaning of the words. There shall be two glosses, one corresponding to the target word, other corresponding to context word. In Lesk algorithm [5] and its variants [6] [7], there has been comparison between different senses of target word and context word in word pairs. In proposed work, the main focus is to decrease the number of comparisons between the word pairs. This would result in performance efficiency in terms of reduced time complexity. A significant improvement has been proved in time efficiency.

II. Related Works

In computational linguistic, Word sense disambiguation (WSD) is the ability to identify the correct meaning of words in context [12]. Contents on Internet are growing rapidly where existing sentences are containing ambiguous words. Removal of ambiguity from sentences that are containing ambiguous words is called word sense disambiguation as in [13]. In global word sense disambiguation, the shotgunWSD is a one of the best algorithm that is unsupervised and knowledge-based. ShotgunWSD has been developed from shotgun sequencing technique that is broadly applied in genome sequencing approach. The ShotgunWSD algorithm applies for word sense disambiguation at document level where it has three phases. The brute force algorithm is applied on short context window in first phase. In second phase, the local sense configurations are assembled by the prefix and suffix matching into the composite configurations where resulting configurations are ranked and sense of each word is detected based on majority voting as on [14]. WSD is a very common problem in the field of natural language processing (NLP). The WSD approaches used till date lies in the following two categories: Knowledge-based approach and corpus-based approach. Knowledge based approaches depends on the availability of knowledge sources such as thesaurus or dictionary or lexical databases (wordnet, BabelNet) to perform disambiguation. Knowledge-based approach uses two types of methods: Overlap method and Graph method. Corpus-based approach uses sense tagged corpus (supervised approach) and sense untagged corpus (unsupervised approach). The first noticeable work of knowledge based approach is by the Lesk [5]. The basic idea used in this algorithm is the sense definition or gloss of the word. In this algorithm, the gloss of the target word is compared with the gloss of all other context words and a score is calculated. Score is defined as the number of common words between the gloss of the target word and the gloss of the context words. Sense with the maximum score is the winner and is assigned as the final sense for the target word in the given context. There are several variants of Lesk algorithm have been proposed [6] [7] [8] [9] [15]. Kilgarriff and Rosenzweig [7] proposed a *Simplified version of Lesk algorithm*. They disambiguated each word individually and it results in the decrease in number of comparison of word pairs. Banerjee and Pederson [6] proposed Adapted Lesk Algorithm for WSD using WordNet. In this algorithm, authors have used the WordNet as lexical resources and they explored the concept of semantic relations defined in the WordNet such as Hypernym, hyponym, meronym, Troponym, Holonym and attributes of each word glosses. In the next work, Banerjee and Pederson [6] presented a new algorithm to measure the semantic relatedness between concepts: *"Extended Gloss Overlaps as a Measure of Semantic Relatedness"*. The measure was number of words matches between the definitions of senses (glosses). They extended the gloss of the concept by incorporating the gloss of other related concepts as defined in the WordNet concept network. A relative evaluation was performed on the variants of Lesk's algorithm by vasilescu et al. [9]. they found that the variants of the Lesk algorithm outperformed the original Lesk algorithm. Baldwin et al. [15] suggested a new algorithm of Machine Readable Dictionary (MRD) based WSD using definition extension and ontology induction. They have used the basic idea of original Lesk [5] and Adapted Lesk algorithm [6]. They experimented over the Hinoki Sense bank and the Japanese Senseval-2 datasets and they found that sense-sensitive definition extension over semantic relations defined in WordNet, integrated with definition extension and word tokenization leads to WSD accuracy above both unsupervised and supervised baselines. Wang and Hirst [16] proposed a method to measure WSD using Naive Bayes similarity. In this method, they replace the overlap mechanism of the Lesk [5] with a general-purpose Naive Bayes model applying the maximum likelihood probability approximate. Brody and Lapata [17] presented an unsupervised approach to determine WSD: Good Neighbours Make Good Senses using distributional similarity. They applied distributional similarity to identify similar words and prepare a sense tagged training dataset without human efforts which is further used to train a standard supervised classifier for doing sense disambiguation. They have adapted Senseval-2 and Senseval-3 dataset for the experiment and got remarkable improvements over state-of-the-art unsupervised methods of WSD. Khapra et al. [18] proposed Bilingual Bootstrapping method for WSD. Considered the bilingual language setup, where the languages under consideration are having fewer amounts of seed data but have the sufficient amount of untagged data. Their idea of tagging the untagged data of one language using the seed data of other language and vice-versa is solely based on bootstrapping method using parameter projection. They use Hindi and Marathi as language pair for their experiment. Khapra et al. [19] proposed a domain specific iterative WSD method for multilingual setting. They considered Hindi, English and Marathi languages for lingual setting. This method is completely dependent on the dominant senses of words (can be nouns, adjectives and adverbs) in the specific domain to accomplish disambiguation. An overall accuracy of 65% on F1-score was reported for all the three languages. Zhong et al. [20] introduced a new WSD method for free text. They utilized the idea of linear support vector machine (SVM) as classifier with some knowledge based features. Singh and Siddiqui [21] proposed an overlapping based WSD method for Hindi. They examined the effect of the removal of stop word, stemming and context window of different sizes and they noticed an improvement of 9.24% and 12.68% in precision and recall respectively. Heyan et al. [1] suggested a new method of unsupervised WSD using collaborative technique. In this work, they utilized the within-sentence relationship (ambiguous sentence) as well as cross sentence relationship (neighbour sentence). The graph-based ranking algorithm is used to perform the disambiguation task. Navigli & Lapata [3] proposed a graph based method for unsupervised WSD. They utilized measures of graph connectivity to find out the most important node (sense) in the graph. They also evaluated the role of lexicon selection and sense inventory as it helps in determining the structure of sub-graph of graph. They used the SemCor dataset for the experiment and show that the degree centrality provides best results compared to the other well known WSD technique such as PageRank, Betweenness Centrality, HITS and Key Player Problem. Basile et al. [22] proposed a WSD algorithm using distributional semantic model. This work relies on the variants [6] [8] of Lesk algorithm. Their approach solely depends on the word similarity function defined over semantic space i.e. they did not use direct matching of words but they used cosine similarity function to get score of overlap. They performed the experiment over SemEval-2013 dataset and adapted BabelNet as lexical database. The Semantic analysis is a crucial part of NLP systems such as information retrieval, data mining, and machine translation. In [23], Authors have discussed about the word vector space model that has been extended to reflect more accurate meaning in context vectors. In [24], Authors have elaborated about the Word Sense Disambiguation in Bengali Language. The Induction technique has been used in first phase of this system where second phase is Word Sense Disambiguation which is developed by the use of Semantic Similarity Measure. ShotgunWSD [25] is a recent algorithm of The Global word sense disambiguation (WSD) which is unsupervised and knowledge-based algorithm. The algorithm has been developed from the Shotgun sequencing technique. The ShotgunWSD contains three phases. The first phase is a brute-force algorithm, the second phase assembles local sense configurations to longer composite configurations and third phase is related to chosen of the sense of each word which is based on a majority voting scheme.

III. Architecture

Initially, the query (in English) is provided by the user in the form of a sentence or a paragraph or a text document. User provided query is an unstructured text (not having any features such as part-of speech, stemmed form of words, etc. attached with it). Each query contains only a single target word (a polysemous word). Target word can be of any type of WordNet word (Noun, Verb, Adjective, and Adverb). A query is provided by the user in English language, must follow the rule of English grammar that is (S + V + O) and must follow the following production rule:

$S \rightarrow NP + VP$

$NP \rightarrow NN \,/\, PRN \,/\, (DET + NN)$

$VP \rightarrow (VB + NP) \,/\, (VB + PP)$

$PP \rightarrow (TO + NP) \,/\, (TO + VP)$

The Subject (S), Verb (V) and Object (O) of any query can be represented as follows,

$S \in NP, V \in VB$

$O \in \{VP - \{VB\}\} \Rightarrow \{O \in NP \,/\, O \in PP\}$

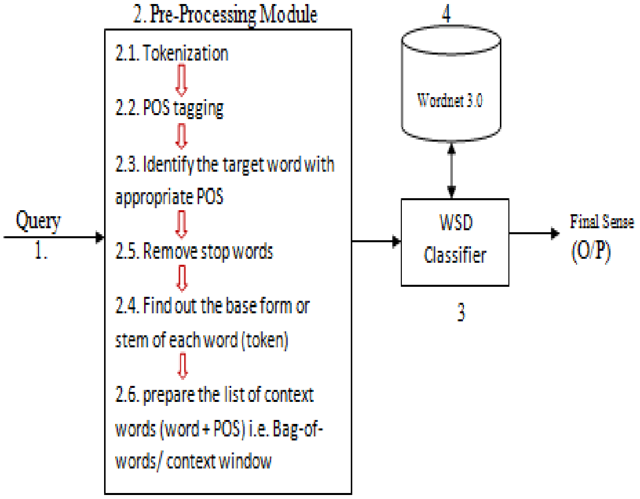

2. Pre-Processing Module
2.1. Tokenization
2.2. POS tagging
2.3. Identify the target word with appropriate POS
2.5. Remove stop words
2.4. Find out the base form or stem of each word (token)
2.6. prepare the list of context words (word + POS) i.e. Bag-of-words/ context window
4
Wordnet 3.0
Query
1.
WSD Classifier
3
Final Sense
(O/P)

Fig 1. Modular Architecture of WSD System.

The architecture of proposed work has been given in Fig. 1. Target word used in the query can be of any type of WordNet word and according to its type context window is prepared which is discussed in following cases:

1) Case 1: w_t can be of any type of WordNet word except verb

In this case, target word can be either noun, Adjective or Adverb. If there are n number of words appears to the left and to the right of the target word then context window is prepared of size (2*n + 1). In the proposed system, only those pairs of words are used that contain target word. So total numbers of words pairs possible are 2n.

2) Case2: when target word is a Verb ($W_T = VB$ & $VB \in VP$)

If the target word is verb and is part of VB in the given query then its dependency is more on the object of the query so left side words are ignored by the proposed system. Total numbers of word pairs formed are n as size of the context window is (n+1).

3) Case3: when target word is a Verb (W_T = VB & VB ∈PP)

If the target word is a verb and it is preceded by a preposition then it is the part of PP. In this case, subject and verb part of VP are ignored.

Identification of WordNet type of the target word is performed by POS tagger. The full discussion of POS tagger is explained in subsection2 of pre-processing module of the proposed system.

A. Pre-Processing Module

Pre-processing module is responsible for taking user query (instance/example containing target word) and converts this query to a structured text. To convert the input query to structured text following steps are performed by the pre-processing module:

1) Tokenization: It is the process of breaking the input query into individual words. Each word is known as token.

2) POS Tagging: This is a process to identify the correct part of speech for each word of the input query. The Adapted standard POS tagger is for annotating the input query. There are many abbreviations are defined to tag words, but some of them have been used that are NN-Noun, NNP-proper Noun, VB - verb, DT - determiners, TO – preposition, etc.

3) Target word is identified and is attached with its proper POS tag.

4) Stop words: the word which appears frequently in the context but the meaning of the context doesn't depend on that word is considered as stop word. In this work, stop word includes all non-WordNet words and auxiliary verbs. If word is a stop word then it will be removed from the context.

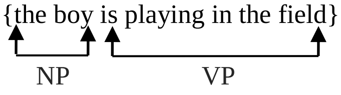
5) Stemming: this is a process of converting each word into its original (base) form.

6) Lastly context window or Bags- of- words is prepared. It contains all the context words including target words.

B. Example:

"The boy is playing in the field."

Tokenization: {The, boy, is, playing, in, the, field}
POS Tagging: {The/DT, boy/NN, is/VBD, playing/VB, in/TO, the/DT, field/NN}
Chunking: {the boy is playing in the field}



Stemming: {The/DT, boy/NN, be/VBD, play/VB, in/TO, the/DT, field/NN}
Stop words: {The, is (be), in}
Context window: {boy/NN, play/VB, field/NN}
After the completion of pre-processing, the bag-of-words contains the word attached with their features like attachment of POS tag. These texts are called structured text that will be delivered to WSD classifier.

C. WSD Classifier

Word sense disambiguation classifier is the last and core module of the proposed system. This module is responsible for the following tasks:

1) Search WordNet: the first task of WSD classifier is to search for the senses of each word $\mathbf{w_i}$ of the context window and retrieve those senses. To retrieve senses, WSD classifier interacts with WordNet which is used as the lexical database. As the type of the word is given in the context window so only matching type of senses are retrieved. For an example, for the word 'play/NN' only noun type of senses is retrieved from the WordNet. After retrieving the senses for all the words of context window, the separate lists of senses have been prepared for the target word and other context words.

2) Score calculation: score is the number of words common between the two glosses. There shall be two glosses, one corresponding to the target word, other corresponding to context word. The methodology for score calculation in given in the algorithm 1.

D. Algorithm

```
Score_calculation ($S_t$ , $S_{cw}$ )
  Input: List of senses definitions (glosses) of Target word and Lists of senses of context words
  Output: Any one sense of Target word
  $SC \leftarrow \Phi$, $SC^C \leftarrow \Phi$, $SC^I \leftarrow \Phi$
    For (i = 1 to | $S_t$ |)
        For (c = 1 to | CW |)
            For ( j = 1 to $N_c^S$)
                $SC \leftarrow SC \cup \{overlap (g_t^i , g_c^j )\}$
            End for
            $SC^C \leftarrow SC^C \cup \{Maximum (SC)\}$
        End for
        $SC^I \leftarrow SC^I \cup \{maximum (SC^C)\}$
    End for
  f = argmax $SC^I$
Return f
```

Some notations are used in the algorithm1 which are explained below:
S_t: List of all sense definitions (glosses) of the target words
S_{cw}: Lists of sense definitions (glosses) of context words.
SC: set of scores calculate for all glosses of any one context word.
SC^C: Set of maximum scores obtained for each context words for ith sense of target word.
SC^I: Sets of maximum score obtained for each sense of target word from all the context words.
f: sense number of the target word which has maximum score and this sense is winner.
Score_calculation () method return the sense number of the target word which having maximum score for all the context words and all other senses of target word. This sense is the correct sense for that context.

IV. ADVANTAGES OF PROPOSED SYSTEM

1) Reduction in Number of Comparison

In the adapted Lesk algorithm, authors have used the concept of overlapping mechanism on the glosses of all the possible pairs of words of the context window where size of the window is equal to the sum of the left and right neighbouring words of the target word and the target word itself. Let 'n' is the number of words to the left and right of the target word then window size is equal to (2*n + 1). In the adapted Lesk algorithm, author have considered all the possible pairs of context words ($^{2n+1}C_2$), but in the proposed work considering only those pairs of words which are having target word. The proposed system is trying to find out the correct sense from the list of senses of the target word, only those pairs can provide the useful information which is containing target word. Total numbers of word pairs possible in the proposed work are 2n.

A/C to adapted Lesk Algorithm,
Total no. of pairs of context window = $^{2n+1}C_2 = (2n+1)*2n/2 = n*(2n+1) \Rightarrow O(n^2)$
But A/C to our approach,
Case1: Total no of pairs of context window = $1 * 2n = 2n \Rightarrow O(n)$
Case2 & case3: Total no. of pairs of context window = $1*n = n \Rightarrow O(n)$
Note: In cases 2 and case 3, the subject part and subject and verb part of the query have been ignored respectively. That means the words have been ignored to the left of the target word so that the size of the window gets decreased to n+1. Total numbers of pairs possible are n.
The proposed work is using less number of word pairs as compared to the adapted Lesk algorithm and so it takes less time to compare the glosses. Word pairs are in O (n) in the proposed work whereas O (n^2) in the adapted Lesk algorithm.

2) Use of POS tagger:

Let,

$S = \{s_1, s_2, s_3, ..., s_i, ..., s_N\}$; set of all senses of the target word '$\mathbf{w_t}$'

Total number of sense = $|S| = N$

$\mathbf{S_{pos} \subset S}$ where pos= {noun, verb}

$S_{noun} = \{s_1, s_2, s_3, ... s_{n1}\}$

$S_{verb} = \{s_1, s_2, s_3, ..., s_{n2}\}$

$|S_{noun}| = n1$, $|S_{verb}| = n2$

Therefore, $|S| = |S_{noun}| + |S_{verb}|$

⇨ $n1+n2 = N$

⇨ $n2=(N-n1)$

$S_i \in S$ is any ith sense of the target word w_t

a) Without using POS tagging:

Let us consider a target word '$\mathbf{w_t}$' in a given context C. Target word should be either Noun or Verb. If this word has total senses available in the WordNet is N. To determine the correct sense of the target word '$\mathbf{w_t}$', now, consider all N senses out of which only one will be the correct sense.

b) Using POS tagging:

To get the correct part-of-speech of the target word, The POS tagger has been used. The target word $\mathbf{w_t}$ must be either a noun or a verb.

Let,

Total sense of $\mathbf{w_t}$ as noun = n1

Total sense of $\mathbf{w_t}$ as verb = (N-n1)

Where n1<=N,

Note: Equality holds if the target word $\mathbf{w_t}$ has any single type of POS tag.

In this case, the type of POS of the target word $\mathbf{w_t}$. If $\mathbf{w_t}$ is a noun then consider only n1 senses otherwise (N-n1) senses to correctly identify the sense of the target word $\mathbf{w_t}$. The lesser number of senses (<N) have been used in both cases except for only one type of POS tag applied for the target word.

POS tagging is applied on all the words of the context window. So it saves a lot of time and space since less numbers of senses are used in the comparison. Pos tagging process can take some extra time but overall it performs better in respect of time. This is analyzed by us by doing several experiments.

Example: The boy is playing in the field.

In the above example, play (stemmed form of playing) is the target word as provided by the user. After POS tagging, the target word play is identified as a verb. So the probability of its meaning's dependency is more on the object of the query.

Query: Q = $\{\underbrace{\textit{the boy}}_{NP}\ \underbrace{\textit{is playing in the field}}_{VP}\}$

NP = {the boy}

VP = {is playing in the field} ⇒ VB = {is playing}; PP = {in the field}

S = the boy, V = is Playing and O = in the field

After stemming and removing stop words final schema of the query is

S = {boy}, V = {play}, O = {field}

case1: If $W_T \neq VB$:

Context window (WC) =

Boy,	play,	field
-1	0	+1

Pairs: {(play, field), (play, boy)}

Case2: If $W_T = VB$

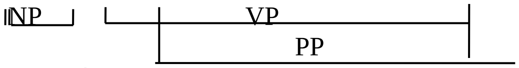
Context window (WC) =

Play,	field
0	+1

Pairs: {(play, field)}

Case3: If $W_T = VB$ and $W_T \in PP$

Query: "*Shyam likes to play with emotions of people*."



NP: {Shyam/NN}

VP: {likes to play with emotions of people}

⇒VB: {likes} && PP: {to **play** with emotions of people}

After removing stop words and stemming,

S = {shyam}, V= {likes}, O = {play with emotions of people}

Context window (WC) = Play, emotion, people
0 +1 +2

Pairs: {(play, emotion), (play, people)}

V. Results and Discussion

The proposed WSD system is implemented on JAVA platform utilizing WordNet3.0 API along with Stanford POS tagger. Lemmatizer has been used to extract the base form of the word. The proposed system has been tested on 50 highly polysemous English words. These 50 polysemous words are taken into three categories - Nouns (20), Verbs (20), and Adjectives (10) and 2000 example sentences are defined for these words. These examples are taken from Wiktionary[1] and other sources of internet[2]. The stop word removal and lemmatization have been implemented which increase the number of overlaps.

The Precision (P), recall (R) and attempt (A) have been measured for the proposed system. The system gives correct output for 1330sentences out of 2000 sentences. The attempt of the system is 100%, so P and R are equal. The P, R, A value are given in Table I. Proposed model classified the query in either correct output or incorrect output. Number of incorrect result is same for both P & R. so P & R are same.

TABLE I. Possible Word Pairs in WSD Approaches

	R	P	A
Lesk	0.23	0.23	100%
A-Lesk	0.47	0.47	100%
M-Lesk	0.665	0.665	100%

Overall performance of the proposed system is better than Adapted Lesk algorithm.

VI. Future Work

In future, the focus will be on the selection of context bag in such a manner which improves the accuracy of system. We can also improve upon the accuracy obtained in case of the single occurrence of target word by working on a standard dataset and thereby comparing the results with that of obtained from other knowledge based approaches. There is also some scope of improvement in WSD as we can use babelNet which is a multilingual WordNet.

VII. Conclusion

In this paper word-sense disambiguation (WSD) problem in natural language processing is studied. WSD governs the process of identifying sense of a word or meaning which is used in a sentence, when the word has multiple meanings (polysemy). The analysis is done and results are compared for both single occurrence of target word as well as multiple (two) occurrences of target words. Experimental results shows that proposed modified lesk method gives 0.665 precision and recall which is higher than Lesk and Adaptive Lesk methods. Due to the improper selection of context bag, lesk and adpative lesk algorithms gives poor results. The proposed method reduces the number of word pair comparisons as compared with the adaptive Lesk Algorithm. Proposed method needs O (n) word pairs as compared with O (n^2) required by the adaptive Lesk algorithm.

References

[1] H. Heyan, Y. Zhizhuo, and J. Ping, "Unsupervised word sense disambiguation using neighborhood knowledge", In 25th Pacific Asia Conference on Language, Information and Computation, pp. 333-342, 2011.

[2] H. Walia, A. Rana and V. Kansal, "A Supervised Approach on Gurmukhi Word Sense Disambiguation Using K-NN Method," 2018 8th International Conference on Cloud Computing, Data Science & Engineering (Confluence), Noida, pp. 743-746, 2018.

[3] R. Navigli, and M. Lapata, "An experimental study of graph connectivity for unsupervised word sense disambiguation", IEEE transactions on pattern analysis and machine intelligence 32, no. 4, pp. 678-692, 2010.

[4] M. Y. Kang, T. H. Min and J. S. Lee, "Sense Space for Word Sense Disambiguation," 2018 IEEE International Conference on Big Data and Smart Computing (BigComp), Shanghai, pp. 669-672, 2018.

[5] M. Lesk, "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 '86, pp. 24-26, New York, NY, USA. ACM, 1986.

[6] S. Banerjee and T. Pedersen. An Adapted Lesk Algorithm for Word Sense Disambiguation Using WordNet. In lexander Gelbukh, editor, Computational Linguistics and Intelligent Text Processing, volume 2276 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, pp. 136–145, 2002.

[7] A.Kilgarriff, and J. Rosenzweig, "Framework and results for English SENSEVAL", Computers and the Humanities, 34(1), pp.15-48, 2000.

[8] S. Banerjee and T. Pedersen, "Extended Gloss Overlaps as a Measure of Semantic Relatedness", Ijcai, vol. 3, pp. 805-810, 2003.

[9] F.Vasilescu, P. Langlais, and G. Lapalme. "Evaluating Variants of the Lesk Approach for Disambiguating Words". In Proceedings of the 4th Conference on Language Resources and Evaluation (LREC), pp. 633–636, 2004.

[10] M. S. Kaysar, M. A. B. Khaled, M. Hasan and M. I. Khan, "Word Sense Disambiguation of Bengali Words using FP-Growth Algorithm," 2019 International Conference on Electrical, Computer and Communication Engineering (ECCE), Cox'sBazar, Bangladesh, pp. 1-5, 2019.

[11] A.Athaiya, D. Modi and G. Pareek, "A Genetic Algorithm Based Approach for Hindi Word Sense Disambiguation," 2018 3rd International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, pp. 11-14, 2018.

[12] R. Navigli, "Word sense disambiguation: A survey", ACM Computing Surveys (CSUR) 41, no. 2, 2009.

[13] E.Faisal, F. Nurifan and R. Sarno, "Word Sense Disambiguation in Bahasa Indonesia Using SVM", 2018 International Seminar on Application for Technology of Information and Communication, Semarang, pp. 239-243, 2018.

[14] M. Butnaru and R. T. Ionescu, "ShotgunWSD 2.0: An Improved Algorithm for Global Word Sense Disambiguation," in IEEE Access, vol. 7, pp. 120961-120975, 2019.

[15] I.T. Baldwin, S. Kim, F. Bond, S. Fujita, and D. Martinez. "A Reexamination of MRD-Based Word Sense Disambiguation", In Journal of ACM Transactions on Asian Language Information Processing (TALIP) Volume 9 Issue 1, Article No. 4, March, 2010.

[16] T.Wang, and G. Hirst, "Applying a Naive Bayes Similarity Measure to Word Sense Disambiguation", In ACL (2), pp. 531-537, 2014.

[17] S.Brody and M. Lapata, "Good Neighbors Make Good Senses: Exploiting Distributional Similarity for Unsupervised WSD", In Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008), Manchester, United Kingdom , Volume 1, pp. 65–72, 2008.

[18] M. Khapra, M. Mitesh, S. Joshi, A. Chatterjee, and P. Bhattacharyya. "Together we can: Bilingual bootstrapping for WSD", In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1, pp. 561-569, 2011.

[19] M. Khapra , P. Bhattacharyya, S. Chauhan, S. Nair, A. Sharma,"Domain specific iterative word sense Disambiguation in a multilingual setting", In Proceedings of International Conference on NLP (ICON 2008), Pune, India, Dec. 2008.

[20] Z. Zhong and H. T. Ng., "It makes sense: A wide-coverage word sense disambiguation system for free text", In Proceedings of the ACL 2010 System Demonstrations, pp. 78-83, 2010.

[21] S. Singh, and T. J. Siddiqui, "Evaluating effect of context window size, stemming and stop word removal on Hindi word sense disambiguation", In Information Retrieval & Knowledge Management (CAMP), IEEE, pp. 1-5, 2012.

[22] P. Basile, A. Caputo, and G. Semeraro, "An Enhanced Lesk Word Sense Disambiguation Algorithm through a Distributional Semantic Model", In COLING, pp. 1591-1600, 2014.

[23] M. Y. Kang,T. H. Min and J. S. Lee, "Sense Space for Word Sense Disambiguation," 2018 IEEE International Conference on Big Data and Smart Computing (BigComp), Shanghai, pp. 669-672, 2018.

[24] A. Sau, T. A. Amin, N. Barman and A. R. Pal, "Word Sense Disambiguation in Bengali Using Sense Induction," 2019 International Conference on Applied Machine Learning (ICAML), Bhubaneswar, India, pp. 170-174, 2019.

[25] A. M. Butnaru and R. T. Ionescu, "ShotgunWSD 2.0: An Improved Algorithm for Global Word Sense Disambiguation," in IEEE Access, vol. 7, pp. 120961-120975, 2019.

Authors' Profile

Manish Kumar received M. Tech in Information Technology from National Institute of Technology, Durgapur, India, in 2015. He is working as an Asst. IT Manager in South Bihar Power Distribution Company Ltd., Bihar, India. His research interest includes Natural Language Processing, Database Management System, Knowledge Management System and Mathematical Analysis.

Prasenjit Mukherjee has 12 years of experience in academics and industry. He was a fulltime Ph.D. Research Scholar in Computer Science and Engineering in the area of Natural Language Processing from National Institute of Technology (NIT), Durgapur, India under the Visvesvaraya PhD Scheme from 2015 to 2019. Presently, He is working as a Data Scientist under Analytics and IT Department, Pune Institute of Business Management, Pune, Maharashtra, India.

Manik Hendre has 5 years of experience in academics and industry. He is a Ph.D. research scholar at the University of Pune. He is currently working as Data Scientist in RamanByte Pune. His research areas include Biometrics Image Processing, Machine learning and Data Analytics.

Dr. Manish Godse has 25 years of experience in academics and industry. He holds Ph.D. from Indian Institute of Technology, Bombay (IITB). He is currently working as an Industry Professor and IT Director in the PIBM, Pune in the area of Artificial Intelligence and Analytics. His research areas of interest include automation, machine learning, natural language processing and business analytics. He has multiple research papers indexed at IEEE, ELSEVIER, etc.

Dr. Baisakhi Chakraborty received the PhD. degree in 2011 from National Institute of Technology, Durgapur, India in Computer Science and Engineering. Her research interest includes knowledge systems, knowledge engineering and management, database systems, data mining, natural language processing and software engineering. She has several research scholars under her guidance. She has more than 60 international publications. She has a decade of industrial and 14 years of academic experience.