www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume – 5 Issue -03 March, 2016 Page No. 15924-15928

Modified Random Walk Algorithm to Improve the Efficiency of Word Sense Disambiguation (WSD)

Rashmi S[1], Hanumanthappa M[2]

[1]Department of Computer Science and Applications,
Bangalore University, Bangalore 560056
rashmi.karthik123@bub.ernet.in

[2] Department of Computer Science and Applications,
Bangalore University, Bangalore 560056
hanu6572@bub.ernet.in

Abstract: ***Natural language processing (NLP) is a field in computer science, artificial intelligence and the linguistics which mainly concentrates on the interactions between human languages (natural language) and the computer. One of the main challenges in NLP is ambiguity. Every language is ambiguous in nature, in the way that one word has multiple meaning and multiple words have same meaning. The ambiguities are generally categorized into two groups: lexical and structural ambiguities. Lexical ambiguity arises where there are two or more possible meaning for a single word. Structural ambiguities appear when a given sentence is interpreted in more than one way due to ambiguous sentence structure. Word Sense Disambiguation (WSD) is defined as the task of finding the correct sense of a word in a specific context. This paper presents our preliminary work towards building WSD system by constructing a corpus. We include a detailed analysis of the factor that affects the WSD algorithm and propose a modified algorithm based on random walk algorithm and compare the working of each of these algorithms***

Keywords: NLP, Lexical ambiguity, Structural ambiguity, WSD, Word Net

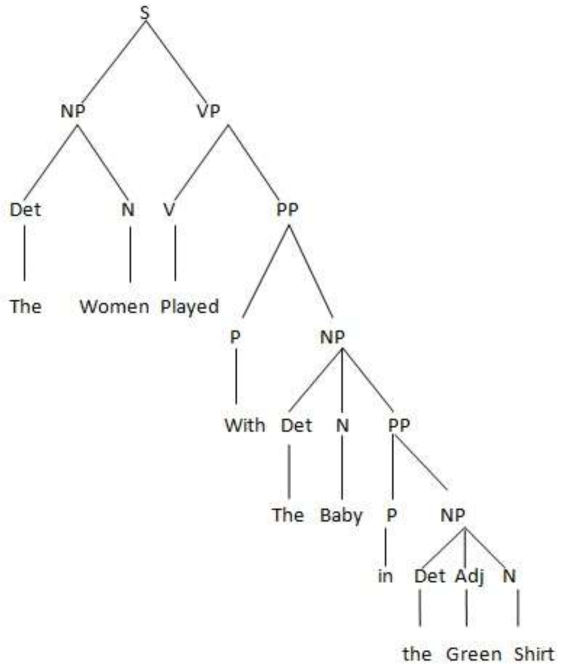
1. Introduction

Lexical ambiguity of a word or phrase pertains of having more than one meaning in the language to which the word belongs. ‘Meaning’ is whatever that is captured by a good dictionary. For instance, the word ‘bank’ has two meaning ‘financial institution’ or ‘edge of the river’. Lexical ambiguity is referred as Homophony and the words which are homophonic in nature are called Homophones. That is, words that are pronounced same but have a different meaning. For ex, Meet and Meat. Lexical ambiguity appears when a lexical item has a surrogate meaning and different part-Of- Speech (POS) tags [1]. Resolving lexical ambiguity is called as WSD. Numerous words have more than meaning, for each word with the multiple we select the meaning which makes the most sense in context. The problem can be resolved by having the list of words and the associated senses. Dictionary or an online resource such as Wordnet can be used. The corpus and the dictionary are the two essential basic resources for processing natural language. The Wordnet is a database for English language. The main aim of Wordnet is to group English words into synonyms and provide semantic relationship between synonym and a short definition.

Wordnet distinguishes between noun, verb, adjectives, and adverbs as all of these follow different grammatical rules. The first task would be to determine the POS for each word. Many words have multiple POS. for ex, consider the word “book”, and observe the following sentences which clearly show the different senses. I) the book is on the table (book here is noun). II) To book a flight (the sense of book here is verb) However due to morphological structure of the verbs and nouns of a sentence are largely disjoint, morphological analysis can resolve some amount of ambiguities. Other kind of ambiguities can be eliminated by rule-based method where we define the rules manually by a classifier [2].

Syntactic ambiguity arises when a sentence have multiple meaning because of the structure of the sentence i.e., its syntax. Notice the following illustration, The women played with the baby in the green shirt. In this example, the baby could be wearing the green shirt or the women could be wearing the green shirt. Below diagram (Figure 1) shows the parsing structure of the sentence mentioned in the above example.

The efficient way to tackle the problem of syntactic ambiguity is rewriting the sentence or by placing the appropriate punctuation wherever necessary.



WordNet: WordNet is a dictionary that stores words and their meaning. Words in the WordNet are arranged in the semantic order rather than alphabetical order. Instead of storing only the meaning of the words, WordNet stores senses, information about POS such as noun, verb, adjective and adverb. WordNet also contains compound words like “financial institution”, “depository financial institution”, “keep one’s eye peeled”. It even holds the information about gloss i.e., small entry explaining of the concept in the synset. In the next section, technical terms of the WordNet are discussed

1.2.1 synonyms and polysemous:

Synonyms are the words with the same sense and meaning. Synonyms are grouped under synonym sets or synsets. Example for synonyms is, beautiful: attractive. Polysemous which are also called as homonymous are those words with the same spelling but different senses. Accident can be the example for polysemous. As “Accident” can mean “mishap” or “anything that happens by chance”

1.2.2 hyponymy and hypernymy:

If synset A is a kind of B, then A is hyponymy (h) and B is called hypernymy. Observe the following example, {glycolic acid} and {sulphuric acid} are the hyponymy as they are all the kind of “Acid” and hence {Acid} can be termed as hypernymy.

1.2.3 holonymy and meronymy:

A is called meronymy of B if A is a part of B and B is called holonymy if B has a part of A. Example, Chapter in a book contains text and other textual matter. Therefore {chapter} is holonymy and {text, textual matter} are meronymy.

2. Methods

The basic approaches to solve ambiguities and adopt WSD are [2]:

Supervised disambiguation

In this method, the system is designed with the examples that are created manually in order to disambiguate the words in the context in which it appears.

The dictionary based or knowledge-based

These systems are treated as the repository of information and also as the source of sense inventory which are used to differentiate the meanings of each word in the context of the sentence. WordNet is considered as the lexical database which necessarily provide meaning of every word along with its complete description such as information about POS, sense and so on

Unsupervised disambiguation

This kind of disambiguation does not depend on the external information sources such as online dictionaries, WordNet and concept hierarchies. There will not be any training set of data to learn from unlike supervised learning. These systems are knowledge-lean.

3 KNOWLEDGE BASED ALGORITHMS

In this section the well known WSD algorithms are explored and examined.

Lesk’s Algorithm:

This was introduced by Michael Lesk in the year 1986. The algorithm is based on the two approaches[3]:

a) When two words are used in close proximity in a sentence, they must be talking of a related topic

b) If one sense of each of the two words can be used to talk of the same topic then the dictionary definitions must use some common words.

Consider the below example which explains the Lesk’s algorithm.

“Pine Cone”, the two words here has its own individual meaning.

a) Pine: “Kinds of evergreen tree with the needle like leaves”

b) Pine: “Waste away through sorrow or illness”.

a) Cone: “Solid body which narrows to a point”.

b) Cone: “something of this shapes whether solid or hallow”.

c) Cone: “Fruit of certain evergreen leaves”.

Each meaning of the word is checked for the similar terms in the meaning of the other word. Here the term “evergreen” is found the meaning of both the words. Hence pine #a= cone #c. Unfortunately this depends on finding the common and identical term between the meanings among various words to disambiguate. And many words in the sentences might not have any related terms between each other. One way to rig this problem is by using the supervised learning which holds the training set of data that consists of a large set of example sentences of the ambiguous words. Each occurrence of the ambiguous word is tagged by a human with the sense in which the word is used. This is well understood by sighting an example.

“Bark of the dog was very loud”. Bark- *sound made by a dog*

“The dog scratched its back on the bark of the tree”. Bark- *covering of the trees.*

In these examples when we use human tagged rules, if the words *dog* and bark appear together and the word *tree* does not, then this indicates that *bark* in this context means *sound made by a dog* and not as *covering of the trees.* Since this approach is human tagging because the sentences are fed by the human and due to this reason it is very tedious in nature. Unsupervised learning uses sources such as online dictionary, WordNet and thesaurus to fetch information

Walkers Algorithm

Walker's algorithm is a thesaurus approach. In the year 1987, Walker proposed an algorithm which is as follows. The algorithm consists of two steps. In the step I, for each sense of the target word find the thesaurus category to which that sense belongs and in the step II, calculate the score for each sense of the context word. A context will add 1 to the score of the sense if the thesaurus category of the word matches that of the sense [4]. Consider the following example. *"The money in the bank fetches an interest of 8% per annum"*. Target word: Bank, Context word: Money, Fetch, Interest, and Annum. According to Walkers' algorithm, the context words add 1 to the sense when the topic of the word matches that of the sense.

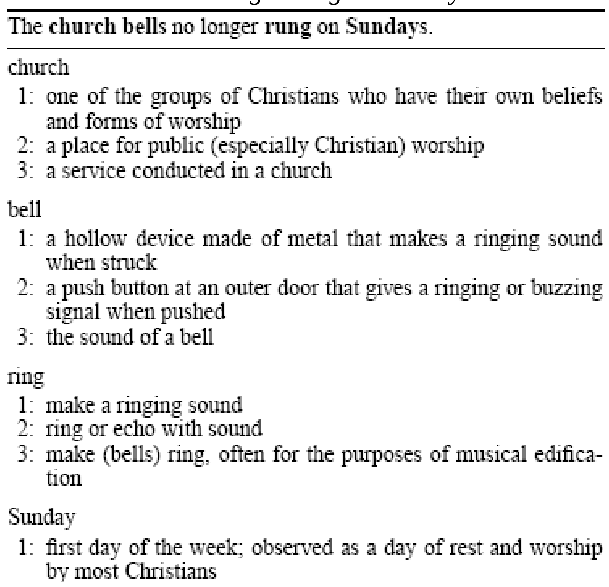
	Sense 1: finance	Sense 2:edge of the river
Money	+1	0
Fetch	0	0
Interest	+1	0
Annum	+1	0
Total	**+3**	0

Table 1- Sense of an example and its weight

WSD using Random Walk Algorithm

Random Walk algorithm is one of the most popular algorithms that is used by search engines for ranking web pages. Example for Random Walk algorithm is PageRank algorithm that is widely used by Google search engine. The algorithm mainly focuses on finding the score of a vertex in the graph which contains variety of senses for a given word. When one vertex links to another vertex it is actually casting a vote for that particular vertex. The algorithm is explained with the below example.

The church bells no longer rung on Sunday



The church bells no longer rung on Sundays.

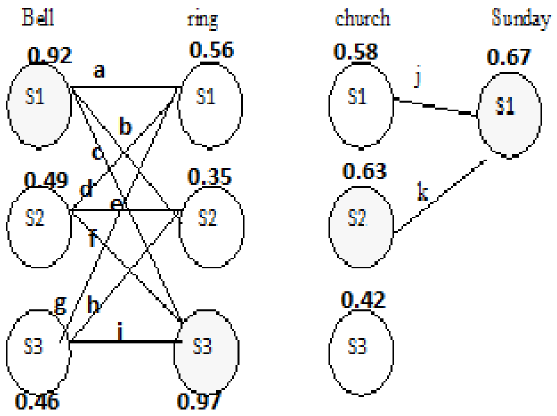
church
1: one of the groups of Christians who have their own beliefs and forms of worship
2: a place for public (especially Christian) worship
3: a service conducted in a church

bell
1: a hollow device made of metal that makes a ringing sound when struck
2: a push button at an outer door that gives a ringing or buzzing signal when pushed
3: the sound of a bell

ring
1: make a ringing sound
2: ring or echo with sound
3: make (bells) ring, often for the purposes of musical edification

Sunday
1: first day of the week; observed as a day of rest and worship by most Christians

Step1: draw vertex for each sense of the target word. **Step 2:** check the association of each sense of every word by applying Lesks' algorithm. **Step 3:** Mark the edges from each vertex. **Step 4:** calculate the term frequency (TF). TF is defined as; number of times a word appears in a document divided by the total number of words in the document. **Step 5:** select the highest TF for each context word. **Step 6:** finally put together all the words obtained from step 5.



4 COMPARISONS OF THE DISCUSSED KB APPROACH ALGORITHMS.

Some of the drawbacks of the above mentioned algorithms are follows. Dictionary based definitions are too small to consider. We will not know the context usage in various scenarios. Here is the table [Table-1] which compares the working of diverse algorithms that are considered

Name of the Algorithm	Accuracy
Lesk's Algorithm	50-60% accuracy even when we consider the 2-word compound statements such as "*pride and prejudice*" "*cat and rats*". However it lacks its efficiency as the algorithm is greatly dependent on the identical terms between the senses of words in the context
Walkers' algorithm	When tested on the normal dictionary an accuracy of 50% was attained. The test was carried out on 10 different highly polysemous English words.
WSD using Random Walk Algorithm	An accuracy of about 37-54% was achieved on SEMCOR corpus. Perhaps this algorithm never explores the relation between various senses of each of the context word

Table-1 Comparison of algorithms

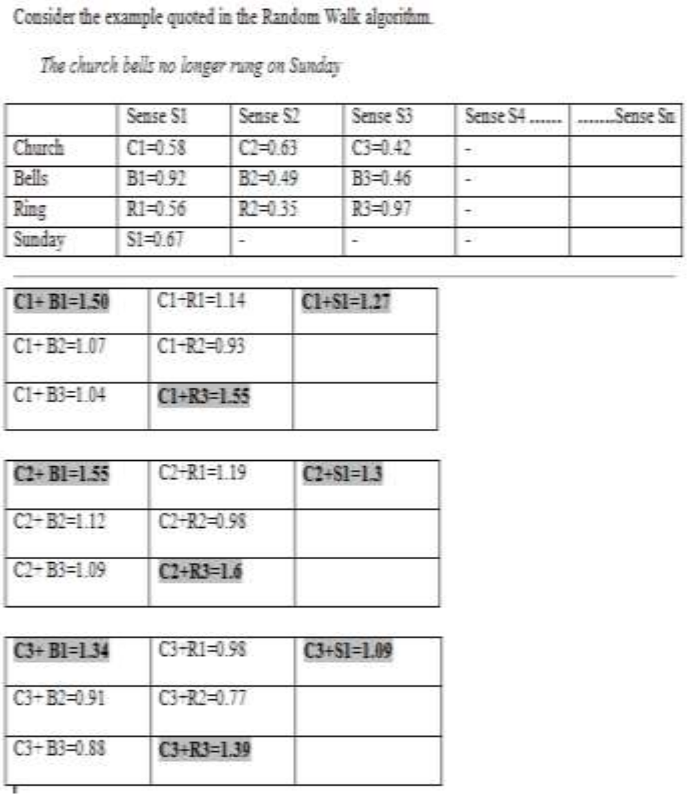
5 MODIFIED VERSION OF RANDOM WALK ALGORITHM

In the Table-1, it is clearly shown that the Random Walk algorithm can achieve the highest accuracy of about 54% when compared with Lesk's and Walkers' algorithms. So keeping this as the bedrock, a new algorithm Modified random walk algorithm is proposed in this section. The biggest drawback of the Random Walk algorithm is that the comparison was not implemented for all the combination of the context word senses. Though the accuracy is 54%, it can be aimed to achieve higher accuracy. In order to overcome this, Modified random walk algorithm concentrates on the association of every possible combination of the senses in the given context word. The Modified random walk algorithm is same till the step 5 of

Random Walk algorithm. The new algorithm is inspired by Random Walk algorithm and also Lesk's algorithm. The Modified random walk algorithm is as follows

Algorithm// Modified random walk algorithm (sentence)

```
//input: sentence which requires WSD
//Output: correct senses for every context word in the given sentence
Step1: Draw vertex for each sense of the target word
Step 2: Check the association of each sense of every word by applying Lesks' algorithm.
Step 3: Mark the edges from each vertex.
Step 4: Calculate the term frequency (TF). TF is defined as; number of times a word appears in a document divided by the total number of words in the document.
Step 5: Select the highest TF for each context word.
Step 5a: For each term frequency of every sense of the context word, prepare a look up table by adding the term frequencies
Step 5b: For each combination of TF, select the highest TF for every context word
Step 5c: Draw a table with these highest TF
Step 5d: For every subset combination of context word senses, select the maximum TF and put this in a separate table
Step 6: Finally put together all the words obtained from the step 5d. This will give you the right sense for every word in the sentence
End
```



Consider the example quoted in the Random Walk algorithm.

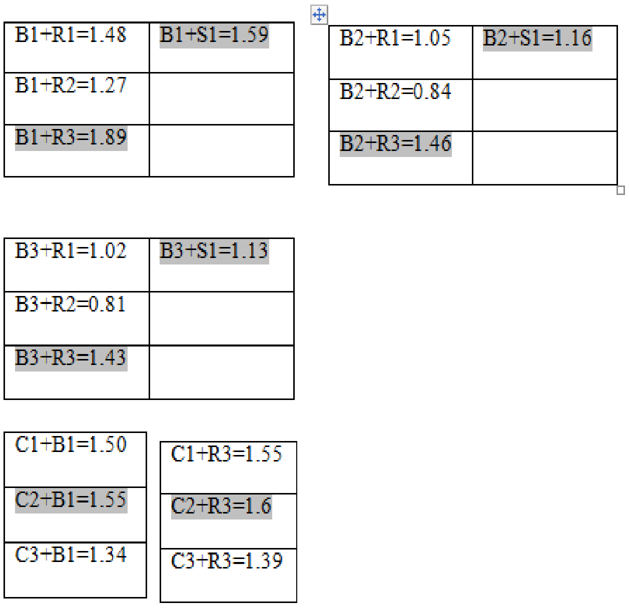
The church bells no longer rung on Sunday

	Sense S1	Sense S2	Sense S3	Sense S4	.......Sense Sn
Church	C1=0.58	C2=0.63	C3=0.42	-	
Bells	B1=0.92	B2=0.49	B3=0.46	-	
Ring	R1=0.56	R2=0.35	R3=0.97	-	
Sunday	S1=0.67	-	-	-	

C1+ B1=1.50	C1+R1=1.14	C1+S1=1.27
C1+ B2=1.07	C1+R2=0.93	
C1+ B3=1.04	C1+R3=1.55	

C2+ B1=1.55	C2+R1=1.19	C2+S1=1.3
C2+ B2=1.12	C2+R2=0.98	
C2+ B3=1.09	C2+R3=1.6	

C3+ B1=1.34	C3+R1=0.98	C3+S1=1.09
C3+ B2=0.91	C3+R2=0.77	
C3+ B3=0.88	C3+R3=1.39	



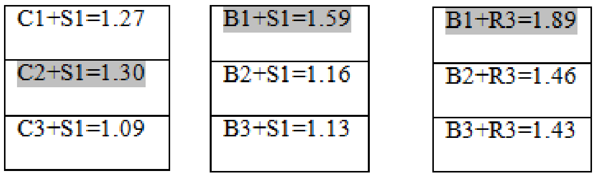
B1+R1=1.48	B1+S1=1.59
B1+R2=1.27	
B1+R3=1.89	

B2+R1=1.05	B2+S1=1.16
B2+R2=0.84	
B2+R3=1.46	

B3+R1=1.02	B3+S1=1.13
B3+R2=0.81	
B3+R3=1.43	

C1+B1=1.50
C2+B1=1.55
C3+B1=1.34

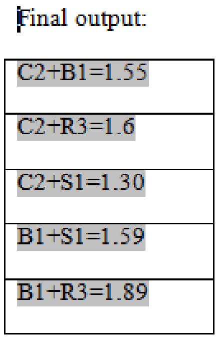
C1+R3=1.55
C2+R3=1.6
C3+R3=1.39



C1+S1=1.27
C2+S1=1.30
C3+S1=1.09

B1+S1=1.59
B2+S1=1.16
B3+S1=1.13

B1+R3=1.89
B2+R3=1.46
B3+R3=1.43



Final output:

C2+B1=1.55
C2+R3=1.6
C2+S1=1.30
B1+S1=1.59
B1+R3=1.89

Therefore the correct senses are

Church: C2 // S2 of *Church* in Random Walk algorithm

Ring: R3 // S3 of *Ring* in Random Walk algorithm

Bell: B1 // S1 of *Bell* in Random Walk algorithm

Sunday: S1 // S1 of *Sunday* in Random Walk algorithm

Conclusion

Natural language is a huge topic of interest. Word Sense Disambiguation (WSD) is a strong subject related focus. It is challenging to achieve because of the syntactic and semantic structures of Natural Language. In this research paper we have studied the approaches that can be adopted to resolve ambiguity in the languages and have proposed a modified WSD algorithm called Modified Random Walk Algorithm.

References:

[1.] Jennifer J Kaplan, "Exploiting lexical ambiguity to help students understand the meaning of random", International Association for Statistical Education, May 2014.

[2.] M. Thangaraj, "Performance Study on Rule-based Classification Techniques across Multiple Database

Relations", International Journal of Applied Information Systems, March 2013

[3.] Pierpaolo Basile, "An Enhanced Lesk Word Sense Disambiguation Algorithm through a Distributional Semantic Model", International Conference on computational linguistics, August 2014

[4.] Walker D. and Amsler R. *The Use of Machine Readable Dictionaries in Sublanguage Analysis in Analyzing Language in Restricted Domains, Grishman and Kittredge (eds)*. LEA Press, pages 69–83, 1986.