Computer Engineering and Intelligent Systems
ISSN 2222-1719 (Paper) ISSN 2222-2863 (Online)
Vol 3, No.8, 2012

A Supervised Word Sense Disambiguation Method Using Ontology and Context Knowledge

Neha R. Kasture* Avinash Agrawal
Department of Computer Science and Engineering, Shri Ramdeobaba College of Engineering and Management, Katol road, Nagpur-440013, India
* E-mail of the corresponding author: nehakasture86@yahoo.co.in

Abstract
Word Sense Disambiguation is one of the basic tasks in Natural language processing. It is the method of selecting the correct sense of the word in the given context. It is applied whenever a semantic understanding of text is needed. In order to disambiguate a word, two resources are necessary: a context in which the word has been used, and some kind of knowledge related to the word. This paper presents a method for word sense disambiguation task using a tree-matching approach. The method requires a context knowledge base containing the corpus of sentences. This paper also gives some preliminary results when a corpus containing the ambiguous words is tested on this system.
Keywords: Natural Language Understanding, Word Sense Disambiguation; Tree-matching; dependent-word matching

1. Introduction

In natural languages a word can have different senses. Humans possess “world knowledge”, which helps them disambiguate words easily. For example, if the word ”bank” appears in a text that also contains words like ”mortgage”, ”assets” or ”bailout”, it is clear that the word ”bank” refers to a financial institution and not to the land near a river. Since computers don’t possess the “world knowledge” used by humans for disambiguation, they need to use other resources for fulfilling this task. A word needs to be disambiguated only if it has multiple senses. Word sense disambiguation (WSD) is the problem of determining in which sense a word having a number of distinct senses is used in a given sentence. There exists several examples of such words where the correct sense of the word with multiple meanings is obvious to a human, but developing algorithms to replicate this human ability is a very tough job. WSD techniques are broadly classified into four categories as proposed in [1].

1.1 Dictionary and Knowledge based Methods

Dictionaries serve two purposes in word sense disambiguation. First, they provide the definitions for the possible senses of a word. Second, the definitions that they provide can be used directly by a disambiguation algorithm. In the early days of WSD, many efforts were put into creating domain specific dictionaries, or micro glossaries [2]. They were based on the idea that, in a specific domain, a word usually has only one sense. While this condition doesn’t hold in every case, the number of ambiguous words is drastically removed in the context of a specific domain [3]. This approach focused on eliminating the possible senses, rather than finding a way to choose from them. Although having some advantages, it had two major drawbacks: they required a lot of manual effort, and, obviously, their applicability was reduced to the domain they were catering for. For example, Lesk disambiguated two words by finding the pair of senses with the greatest word overlap in their dictionary definitions [4]. Overlap was calculated by simply counting the number of words that two definitions have in common.

1.2 Supervised Methods

These methods make use of context to disambiguate the words. Supervised method includes a training phase and a testing phase. In training phase a sense annotated corpus is required, from which syntactic and semantic features are used to create a classifier. In the testing phase the word is classified into senses. These methods often suffer due to data scarcity problem and it is equally hard to acquire sufficient contextual information about senses of large number of words in natural languages. The most popular classifier used for word sense disambiguation is the Naive Bayes classifier, introduced by Gale et al. [5]. Other popular techniques are Decision Lists [6], instance-based classifiers [7] or neural network [8]. Recently, Support Vector Machines have also been used for WSD classifiers [9].

1.3 Semi- supervised Methods

These methods make use of small annotated corpus as seed data in bootstrapping process as proposed in [10].

1.4 Unsupervised Methods

These methods perform word sense disambiguation without using any resource. However, these algorithms don’t perform strict sense tagging, which is to assign one of several possible meanings to an instance of a word. Instead, this class of algorithms performs sense discrimination: they group instances of a word by meaning. They cannot however; label these groups, since the possible senses and their definitions are not available. Therefore, unsupervised WSD is similar to the clustering problem. Due to their nature, unsupervised WSD techniques are not as widespread as other techniques. Nevertheless, several clustering algorithms have been adapted for solving this problem [11].

Disambiguation accuracy largely depends on the number of words to be disambiguated. Limited words can be disambiguated with comparatively smaller knowledge base but as the number of words increase the knowledge base to disambiguate it should also increase proportionally. The main challenging part is thus to overcome this deficiency in context knowledge. It is very important to maintain a comprehensive, dynamic and up-to-date data in the knowledge base. Human beings can keep themselves updated with latest words or new vocabulary but to maintain same level of maintenance in updating latest knowledge for a machine needs continuous effort.

The paper is structured as follows. Section II is the related work in this particular method for disambiguating a word. Section III gives the details about the methodology of the disambiguation system. It describes the representation of Tree-bank and Gloss-bank required for the Tree-matching procedure. Section IV focuses on the implementation details of the system and its working. Section V shows the results of the testing performed on the system. Finally, the last section concludes the work done and considers its future aspects.

2. Related Work

Various graph based methods and different models gained lot of attention for word sense disambiguation tasks. Graph-based methods used for Word Sense Disambiguation are mostly unsupervised methods that rely on the lexical Knowledge Based graph structure for concluding the relevance of word senses in the given context. Usually the reference resource used in majority is WordNet [12]. In most of the graph-based methods each context word of the target word is represented as a vertex. If two vertices co-occur in one or more instances then they are connected via an edge. After a co-occurrence graph is made, different graph clustering algorithms can be applied to partition the graph. Each cluster then represents the set of words which are semantically related to a particular sense.

Many natural language processing tasks consists of labeling sequences of words with linguistic annotations, e.g. word sense disambiguation, part-of-speech tagging, named entity recognition, and others. A graph based sequence data labeling algorithm is presented in [13] as solution for such natural language annotation tasks. The algorithm simultaneously annotates all the words in sequence by exploiting relations identified among word labels, using random walks on graphs encoding label dependencies. The basic idea behind this algorithm is of "recommendation" or "voting". When one vertex links to the other, it is basically casting a vote for that other vertex. More the number of votes cast by the vertex, higher the importance of the vertex.

Klapaftis and Manandhar (2008) proposed a graph-based approach as in [14] that represents pair of words as vertices instead of single words. The idea behind the approach was that single words can appear with more than one senses of the target word. They hypothesized that the pair of words is unambiguous. This approach achieved good results in both evaluation settings of SemEval-2007 task. If one or more pairs of words representing the induced sense co-occur in the test instance then it is disambiguated towards one of the induced senses. But since co-occurrence of a pair of words is less likely than the occurrence of a single word, this approach creates a data scarcity problem.

Korkontzelos and Manandhar [15] presented an unsupervised graph based method for word sense induction and disambiguation. They relaxed the condition imposed by the previous approach by allowing assignment of either a word or word pair to each vertex of the graph. This could be done because in some cases a single word is unambiguous. If the word is found to be unambiguous then it is used as a single word vertex. Otherwise, it is represented as pair-of-word vertex. Word senses are induced by clustering the constructed graph. While disambiguating the word, each induced cluster is scored according to number of its vertices found in the context of the target word. Thus, this system works in two major steps, word sense induction and then disambiguation.

Klapaftis and Manandhar [16] proposed that graphs often exhibit the hierarchical structure that goes beyond simple flat clustering. They present an unsupervised method for inferring the hierarchical grouping of senses of a polysemous word. The inferred hierarchical structures (binary trees) are then applied to word sense disambiguation. In this method the vertices of the graph are contexts of the polysemous word and edges represent the similarity between contexts. The binary tree produced by this method groups the context of polysemous word at different heights of the tree and thus, induces the word sense at different levels of sense granularity.

Diego De Cao et al. [17] presented an adaptation of Page Rank algorithm for Word Sense Disambiguation.

It preserves the reachable accuracy while significantly reducing the processing time. They exploit distributional evidence that can be automatically acquired from corpus, to amplify the performance of sentence oriented version.

This type of algorithm has achieved improved efficiency and a speed-up of two orders of magnitude at no cost in accuracy. They employed Senseval 2007 coarse WSD dataset to measure the accuracy. Their evaluation is based on two main aspects. First the impact of topical expansion at sentence level on the accuracy reachable by Personalized Page Rank PPR. Second analysis of the efficiency of algorithm and its impact on the sentence or word oriented perspective. In order to validate the hypothesis that Latent Semantic Analysis (LSA) is helpful to improve time complexity of WSD, analysis of processing times of different data sets was conducted, to compare methods and resources used.

Ping Chen et al. [19] presented a fully unsupervised word sense disambiguation method that requires only a dictionary and un-annotated text as input. It is a completely automatic approach that overcomes the problem of brittleness suffered in many existing methods and makes broad-coverage word sense disambiguation feasible in practice. The context knowledge base is built by taking huge amount of semantically correct sentences from electronic text collection of books and numerous web documents. Such type of data provides broad and up-to-date context knowledge necessary for word sense disambiguation process. To parse the sentences into parsing trees dependency parser Minipar is used and then the trees are saved in files. The WSD approach suggested is very insightful which says that "If a word is semantically coherent with its context then at least one sense of this word is semantically coherent with its context". They have evaluated this approach with SemEval-2007 Task 7 (Coarse-grained English All-words Task) data set, and have achieved F-scores approaching the top performing supervised WSD systems. In addition to the unsupervised approach they have also suggested different matching strategies like dependency matching, dependency and backward matching, dependency and synonym backward matching, dependency and synonym dependency and finally dependency, backward, synonym backward and synonym dependency matching to improve quality of disambiguation. This work is the main motivation behind the experiments performed in this paper and the matching style followed is dependency and backward matching. The method suggested in this paper makes use of concept of similarity matrix to evaluate the score of different senses as shown in Section III.

3. Methodology

The basic purpose of the proposed system is to disambiguate the correct sense of the specified ambiguous word in the input sentence given to the system. The input to the system is given as a parse tree of the sentence containing the ambiguous word. The objective of the system is to disambiguate the target word by using the tree-matching algorithm. The algorithm matches the tree of input sentence with the trees of all the glosses of the target word available in the database of the system one by one. The trees of different glosses of the ambiguous word should be available with the system to perform tree-matching. Thus, the main objective of the system is to perform tree-matching between tree of input sentence and tree of each gloss of ambiguous word. While performing the matching with gloss the dependent words of the words occurring in the gloss-tree are also considered. These dependent words are taken from the knowledge base which is the corpus of sentences containing various ambiguous words used in different contexts. The algorithm generates a score value for each gloss and the gloss which gives maximum score should be selected as the correct sense of the ambiguous word in the context of the sentence. Figure 1 shows the block diagram of the proposed system.

3.1 Knowledge Base Creation

This is a very basic step which aims to collect as many semantically and grammatically correct sentences as possible. The objective of this system is to find the instances of sentences containing the ambiguous words. These instances provide the source of dependent words which usually co-occur with the ambiguous words. These dependent words are used while performing the tree matching. Many sources can be considered while constructing the knowledge base like Electronic text collections which include thousands of books mostly written by professionals. Such sources give many valid sentences covering wide variety of words. Another source for dynamic text collection is web documents. It is an ideal source for broad coverage and up-to-date context knowledge which is necessary for WSD.

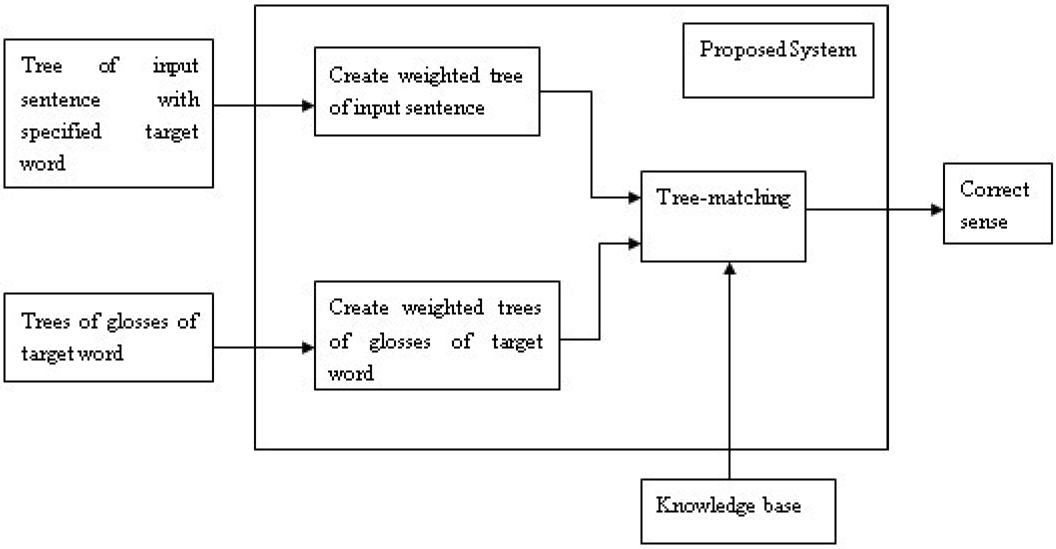

Figure 1. Block diagram of the proposed system

Following two major text sources considered in this system.

3.1.1 Text from Travelogue of Internet

A part of the sentences considered in preparing the knowledge base are taken from the Internet. Travelogue on Internet gives wide variety of sentences used in different contexts.

3.1.2 Text from Newspaper

The second source of text is sentences from newspapers. These sentences ensure coverage of words in different contexts as well as their latest usage. Acquisition of sentences from these two sources ensures variety of word usages in text and hence, strengthens the knowledge base. The size and quality of knowledge base positively impacts the disambiguation quality.

3.2 Processing the input sentence containing the ambiguous word

The system accepts the input sentence containing the ambiguous word in tree form. Once the tree of input sentence is available the system gives weights to each node of the tree. Following steps explain the processing done on the input sentence.

3.2.1 Parsing the sentence into a tree

Dependency parsers are used for parsing the text sentences. Dependency Parsers are widely used in Computational Linguistics and natural language processing. Parsing the sentence into trees actually helps us to establish dependency relations between the words. An example is shown in Figure 2, in which the sentence "*Audience watches the swaying minarets*" is parsed into a tree. In a dependency relation "$word_i \rightarrow word_j$", $word_i$ is the main word and $word_j$ is its dependent word.

3.2.2 Creating Weighted Parse tree of an input sentence

Once the tree of input sentence is available to the system, weights are assigned to the nodes of the tree. The weight is given as the reciprocal of distance from the target word. For example the weighted parse tree of sentence "*Audience watches the swaying minarets*" in which "watches" is the target ambiguous word will be created as shown in Figure 3.

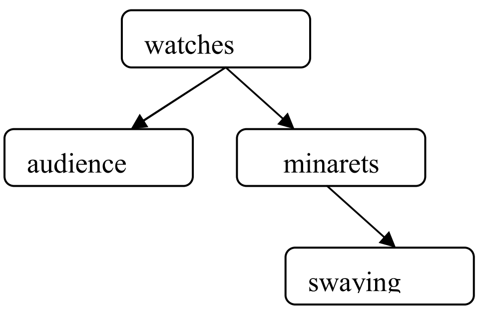

Figure 2. Parse tree of an input sentence

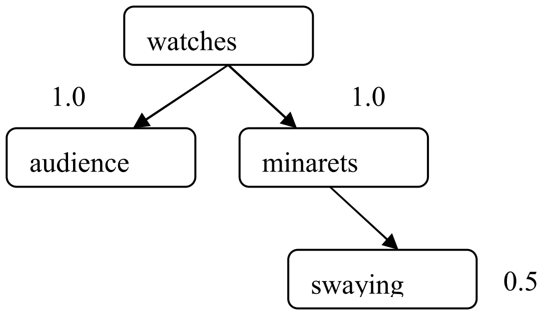

Figure 3. Weighted Parse tree of an input sentence

3.3 Processing the glosses of the ambiguous word

For every ambiguous word in the Tree-bank of Knowledge base, its glosses are taken from the WordNet. Since ambiguous words have multiple meanings, those text sentences of different meanings are taken from the WordNet. In this experiment, instead of taking all the glosses for every ambiguous word from the WordNet, two or three or in some cases four different meanings of the ambiguous word are considered to test the accuracy of the method. Following processing is to be done on the glosses.

3.3.1 Parsing the glosses into a tree

The glosses of the ambiguous words should be available to the system in the form of a tree. Dependency parser is used to parse the glosses of the ambiguous words. The logic of creation of tree is same as used for parsing the input sentence containing the ambiguous word but in this case the root node is always taken as the ambiguous word itself.

Let us consider the following two glosses of the ambiguous word "watches".

- Look attentively
- A small portable timepiece

Figure 4 shows the tree representation of the two glosses of the word "watches".

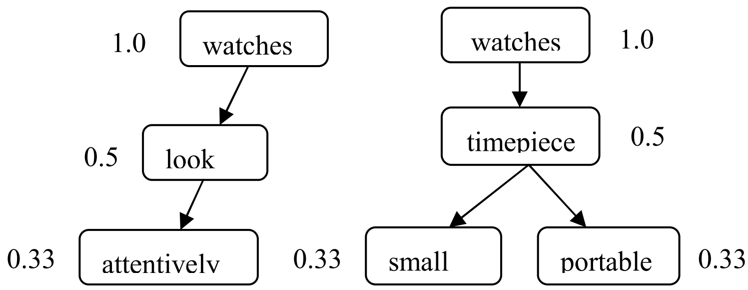

Figure 4. Weighted Gloss tree of "watches"

3.4Similarity Matrix to measure Semantic Relevancy

The Knowledge base consists of corpus of sentences containing the ambiguous words as well as many other words which co-occur with the ambiguous words. Similarity matrix measures the semantic relevancy between different words occurring in the knowledge base. This matrix gives the probability of each word with its co-occurring word. The words are said to be co-occurring if they appear in the same line in the knowledge base. This semantic coherence is a very important factor while choosing between different senses of the ambiguous word. Also, the similarity matrix is a dynamic entity, if the sentences in the knowledge base increases the probabilities also change correspondingly.

3.5Proposed Tree-Matching method

The following is the proposed tree-matching method for satisfying the objective of disambiguating the target word. The weighted trees of both input sentence as well as the glosses of ambiguous words and the probabilities of co-occurring words in the knowledge base should be available with the system while performing the tree-matching.

Let S be the input sentence containing the specified ambiguous word w.

Input:

- Weighted tree T_s of input sentence S containing the target word w.
- Weighted gloss trees T_{wi} for each gloss w_i of ambiguous word available in the database of the system.

- Probabilities from Similarity matrix.

Algorithm:

- Let w_{si} be the weight of each word in the input sentence which is given as the reciprocal of the distance from the ambiguous target word.
- Let w_{wi} be the weight of the node in gloss tree T_{wi} which is given as the reciprocal of the level number.

1. For each gloss w_i of ambiguous word w
   a. Get the gloss tree T_{wi} ;
   b. For each node n_{wi} in the gloss tree T_{wi} of the ambiguous word w
      i. Load the dependent co-occurring words D_{wi} from knowledge base ;
      ii. For each node n_{sj} in the tree T_s of input sentence
         - If n_{sj} belongs to any of the words in the set of words D_{wi}
            - Extract the probability of co-occurrence p_{ji} between n_{sj} and n_{wi} ;
            - Score= Score + w_{si} * w_{wi} * p_{ji} ;
2. Choose the gloss with highest score as the correct sense of ambiguous word.

4. Implementation

The Tree-Matching algorithm proposed disambiguates the ambiguous word specified by the user in the input sentence. Let us consider an example to demonstrate the complete implementation of the proposed system. Suppose disambiguation of the word "book" in the sentence "Book a case against corrupt politicians" is to be performed. As per the proposed method the input to the system should be in the form of trees. Thus the desired input is first given to the system.

4.1Input to the system

The weighted tree of the input sentence and the weighted gloss trees of the ambiguous word "book" are given as an input to the system.

4.1.1 Weighted tree of an input sentence

Figure 5 shows the weighted tree of sentence "Book a case against corrupt politicians". The weights are assigned to each node of the input sentence as the reciprocal of the distance from the target ambiguous word.

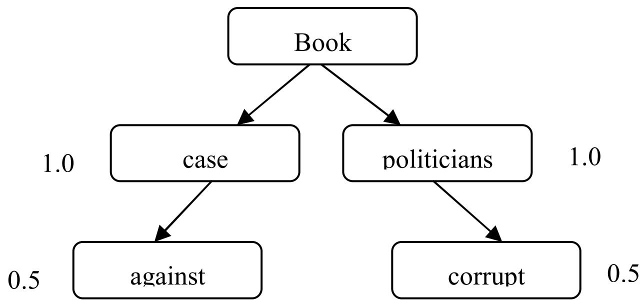

Figure 5. Weighted parse tree of an input sentence

4.1.2 Weighted trees of glosses of an ambiguous word

Figure 6 shows the weighted trees of two glosses of the bank present in the database. Two glosses considered in this example are:

A written work or composition that has been published

Record a charge in police register

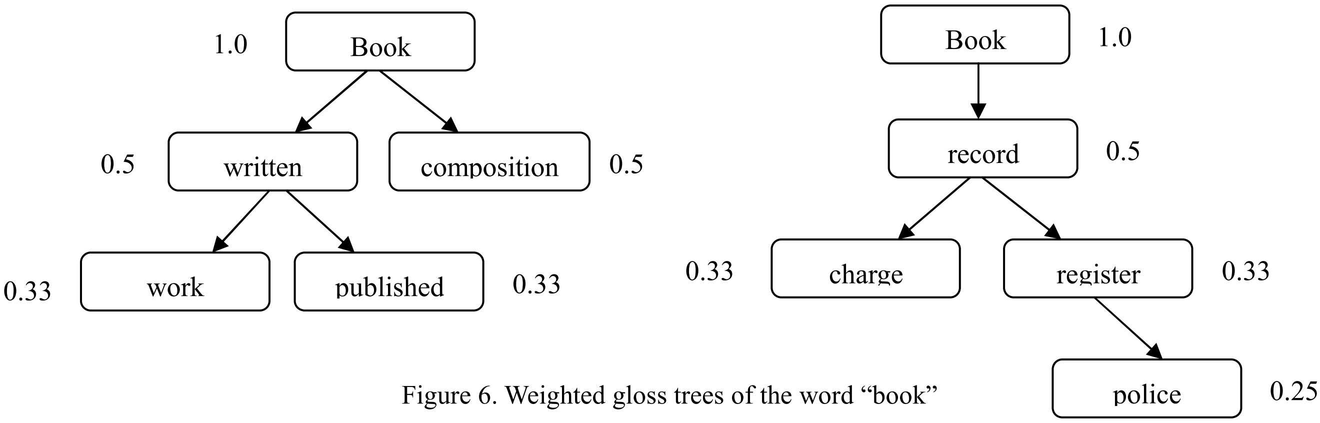

Figure 6. Weighted gloss trees of the word "book"

4.2Implementation of the proposed Tree-Matching method

The steps of the Tree-Matching method described earlier are then applied to disambiguate the word “book” in the sentence “Book a case against corrupt politicians”. For every gloss the score is calculated and the highest score i.e., the gloss most semantically coherent with the context of the sentence is selected as the correct sense.

4.1.2 Score Calculation

Gloss-1: A written work or composition that has been published

- The dependent words for every word in the gloss are extracted.
- From the set of dependent words extracted we get dependent words (against, politicians) from the knowledge base for the word “book” which matches with the words of input sentence.
- No other word of first gloss tree contains dependent words that match with the input sentence.
- Probability of co-occurrence is taken from similarity matrix between the word pair “against” and “book” and “politicians” and “book’.
- Thus, the score for first gloss can be calculated as:

0.5 * 1 * 0.7544 + 1.0 * 1.0 * 0.3772 = 0.7544

- The ceil value of score is taken.
- Thus, Score=1

Gloss-2: Record a charge in police register

- The dependent words for every word in the gloss are extracted.
- For the second gloss, we get the following dependent words for each word of gloss that match with input sentence.
  - Book = (against, politicians)
  - Charge =(against, corrupt, politicians)
  - Register = (politician, against, corrupt)
  - Police = (against, politician)
- No other word of first gloss tree contains dependent words that match with the input sentence.
- Probability of co-occurrence is taken from similarity matrix between each of the above word from gloss and its dependent words found from knowledge base.
- Thus, the score for second gloss can be calculated as:

0.5 * 1 * 0.7544 + 1.0 * 1.0 * 0.3772
+ 0.5 * 0.33 * 0.7544 + 0.5 * 0.33 * 0.3772 + 1 *0.33* 0.3772
+ 1 * 0.33* 0.7544 + 0.5 * 0.33 * 0.7544 + 0.5 *0.33 * 0.3772
+ 0.5 * 0.25 * 0.7544 + 1 * 0.25 * 0.3772 = 1.689

- The ceil value of score is taken.
- Thus, Score=2

As per the steps of proposed method, the maximum score is selected as the correct sense of the target word. Hence, the correct meaning of the word “book” in the input sentence will be “Record a charge in police register”.

5. Experimental Results

The system is tested by three different experiments. A Test corpus of 1000 random sentences containing the ambiguous words from the knowledge base is created. Some sentences are taken from people not involved in the development work and some are taken from the newspapers. The 1000 sentences of Test corpus contain instances of different senses of about 350 ambiguous words from knowledge base. The test corpus is tested thrice, first when knowledge base contains 1000 sentences then 1500 sentences and finally 2000 sentences. Testing the same set of random sentences over different strengths of knowledge base gives the idea of disambiguation quality when magnitude of knowledge base increases. The results of testing show that as the strength of knowledge base increases the disambiguation quality also increases. The system does not give any output if the target word is not found in the gloss tree-bank or if its sufficient knowledge in the form of dependent words is not present in the tree-bank of knowledge base. The result shows the impact of magnitude of context knowledge on the disambiguation of target words. Natural languages are dynamic in nature, with time new words emerge and usages of words tend to change. Human beings can keep themselves updated with latest words or new vocabulary but to maintain same level of maintenance in updating latest knowledge for a machine needs continuous effort. Hence, if the domain or range of vocabulary is not limited the performance of system drops in practice. Figure 7 shows the graphical representation of precision, recall and F-measure.

Table 1. Disambiguation Results

Knowledge Base	Test Corpus	Precision	Recall	F-measure
1000	1000	98.14	53.53	69.26
1500	1000	98.70	76.76	86.35
2000	1000	98.87	88.88	93.60

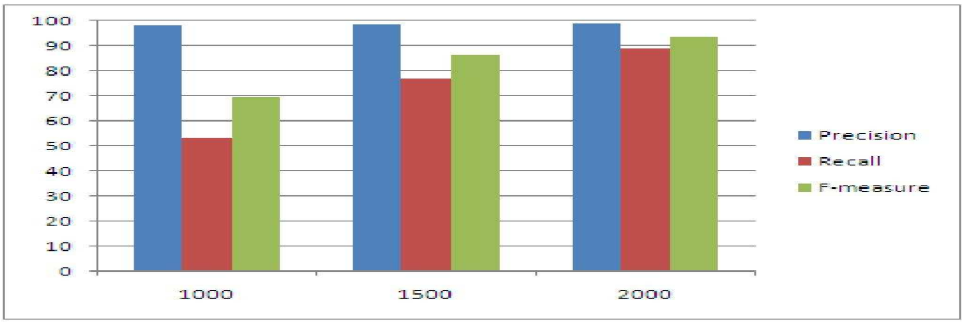

Figure 7. Graphical representation of the result

6. Conclusion and Future work

This paper proposed a word sense disambiguation method based on the Tree-matching approach. The results of the experiments clearly reflect that the system works well in disambiguating the words currently considered in comparatively smaller knowledge base and the disambiguation performance can be improved further by collecting and incorporating more context knowledge. There is also a scope to strengthen the knowledge base with more current, up-to-date vocabulary of words to make the system dynamic. The future work includes:

- Continue building a knowledge base from different sources of text to include variety of sentences from multiple domains to enlarge the scope of disambiguation.
- There is also a scope to strengthen the knowledge base and system with more current, up-to-date vocabulary of words and their glosses to make the system dynamic.
- Attempting to design matching strategy with additional parameters to give improved performance over present matching.

References

[1] Agirre, Eneko, Philip Edmonds (eds.). 2006. Word Sense Disambiguation: Algorithms and Applications, Springer.

[2] Nancy Ide and Jean V´eronis. Introduction to the special issue on word sense disambiguation: the state of the art. Comput. Linguist., 24(1):2–40, 1998.

[3] Warren Weaver. Translation. In Mimeographed, pages 15–23. MIT Press, 1949. Proceedings of Fifth Annual International Conference on Systems Documentation (Toronto, Ontario, Canada). V. DeBuys, Ed. SIGDOC '86.

[4] Lesk, M. 1986. Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice-cream cone. In the Proceedings of Fifth Annual International Conference on Systems Documentation (Toronto, Ontario, Canada). V. DeBuys, Ed. SIGDOC '86.

[5] William A. Gale, Kenneth W. Church, and David Yarowsky. A method for disambiguating word senses in a large corpus. Computers and the Humanties, 26:415–439, 1992.

[6] David Yarowsky. Decision lists for lexical ambiguity resolution: application to accent restoration in Spanish and french. In Proceedings of the 32nd annual meeting on Association for Computational Linguistics, pages 88–95, Morristown, NJ, USA, 1994. Association for Computational Linguistics.

[7] Hwee Tou Ng and Hian Beng Lee. Integrating multiple knowledge sources to disambiguate word sense: an exemplar-based approach. In Proceedings of the 34th annual meeting on Association for Computational Linguistics, pages 40–47, Morristown, NJ, USA, 1996. Association for Computational Linguistics.

[8] Geoffrey Towell and Ellen M. Voorhees. Disambiguating highly ambiguous words. Comput. Linguist., 24(1):125–145, 1998.

[9] Clara Cabezas Philip, Philip Resnik, and Jessica Stevens. Supervised sense tagging using support vector machines, 2001.

[10] Hearst, M. (1991) Noun Homograph Disambiguation using Local Context in Large Text Corpora, Proc. 7th Annual Conference of the University of Waterloo Center for the new OED and Text Research, Oxford.

[11] Hinrich Sch¨utze. Automatic word sense discrimination. Comput. Linguist., 24(1):97–123, 1998.

[12] G. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. Miller. 1990. An on-line lexical database. *International Journal of Lexicography*, 13(4):235-312.

[13] Rada Mihalcea. Unsupervised large-vocabulary word sense disambiguation with graph-based algorithms for sequence data labeling. In *Proceedings of Human Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing,* pages 411-418, Vancouver, British Columbia, Canada, October 2005. Association for Computer Linguistics.

[14] I. Klapaftis and S. Manandhar. 2008. Word sense induction using graphs of collocations. In *proceedings of EACI-2008*, Patras, Greece.

[15] I. Korkontzelos and S. Manandhar. 2010. UoY: Graphs of Unambiguous Vertices for Word Sense Induction and Disambiguation. In Proceedings of 5th International Workshop on Semantic Evaluation, ACL 2010,pages 355-358, Uppsala, Sweden.

[16] I. Klapaftis and S. Manandhar. 2010. Word Sense Induction and Disambiguation Using Hierarchial Random Graphs. In the Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, pages 745-755, MIT, Massachusetts, USA.

[17] Diego De Cao, Roberto Basili, Matteo Luciani, Francesco Mesiano, Ricardo Rossi. 2010. Robust and Efficient Page Rank for Word Sense Disambiguation. In the proceedings of 2010 Workshop on Graph-based Methods for Natural Language Processing, ACL 2010, pages 24-32, Uppsala, Sweden.

[18] J. Veronis. 2004. Hyperlex: lexical cartography for information retrieval. *Computer Speech & Language*, 18(3):223-252, July.

[19] Ping Chen, Wei Ding, Chris Bowes, David Brown. 2009. A Fully Unsupervised Word Sense Disambiguation Method Using Dependency Knowledge. In the *Proceedings of the 2009 Annual Conference of the North American Chapter of the ACL*, pages 28-36, Boulder, Colorado.