IOSR Journal of Engineering (IOSRJEN) **www.iosrjen.org**
ISSN (e): 2250-3021, ISSN (p): 2278-8719
Vol. 09, Issue 5 (May. 2019), ||S (IX) || PP 80-85

Supervised Approach to Word Sense Disambiguation

Aditi Salodkar [1], Mrunali Nagwanshi [2], Ms.Bhavana Gopchandani[3]
[1] *Department of Computer Science Engineering , Jhulelal Institute of Technology, RTMNU, Nagpur, India*
Corresponding Author: Aditi Salodkar

Abstract*: In natural language processing, there exist a problem of determining which "sense" (meaning) of a word is activated by the use of the word in a particular context. Natural language is ambiguous. It is an easy task for a human to understand and disambiguate the ambiguous words but a trivial task for a computer to do so. Ambiguity can occur at various levels of NLP- lexical, syntactic, and semantic and discourse level. The project here concentrates on Lexical Semantic ambiguity task. Lexical Semantic ambiguity takes place when a word/lexicon or a phrase has multiple meanings associated with it. Word Sense Disambiguation (WSD) aims to disambiguate the words which have multiple sense in a context automatically. Sense denotes the meaning of a word and the words which have various meanings in a context are referred as ambiguous words. It is the task of understanding the sense of an ambiguous word in a piece of context. It basically assigns the appropriate sense to a word depending on the particular context where it occurs in an automated manner. In this report, we propose supervised Machine Learning approach for Word Sense Disambiguation task in English language.*

Keywords: *Ambiguous word,, Lexical Semantic, Natural Language Toolkit(NLTK),, Supervised Machine Learning* **.**

I. Introduction

Word sense disambiguation has been implemented in many Indian languages like Assamese, Manipuri, Tamil, Malayalam, Hindi, Kannada, Nepali, and Punjabi using various approaches like Supervised ,Knowledge-based, Unsupervised and semi-supervised approaches. It is necessary that WSD is to be implemented in English language. Ambiguity can occur at various levels of NLP- lexical, syntactic, and semantic and discourse level. The project here concentrates on Lexical Semantic ambiguity task. For example, consider the two sentences. "The bank will not be accepting cash on Saturdays." And "The river overflowed the bank." The word bank in the first sentence refers to the commercial (finance) banks, while in second sentence, it refers to the river bank. It is easy for humans to understand the meaning of "bank" word by understanding logic behind sentence. But difficult for machine to understand actual meaning of such ambiguous words. The ambiguity that arises due to this, is tough for a machine to detect and resolve. WSD is implemented by using supervised approach. This application enable the user to find exact meaning of ambiguous word in a sentence. WSD is the task of understanding the sense of an ambiguous word in a piece of context. It basically assigns the appropriate sense to a word depending on the particular context where it occurs in an automated manner.

II. Literature Survey

1.Ariel Raviv and Shaul Markovitch introduce Concept-Based Disambiguation (CBD),[1] a novel framework that utilizes recent semantic analysis techniques to represent both the context of the word and its senses in a high-dimensional space of natural concepts. The concepts are retrieved from a vast encyclopedic resource, thus enriching the disambiguation process with large amounts of domain-specific knowledge. In such concept-based spaces, more comprehensive measures can be applied in order to pick the right sense. Additionally, they introduce a novel representation scheme, denoted anchored representation, that builds a more specific text representation associated with an anchoring word. We evaluate our framework and show that the anchored representation is more suitable to the task of word sense disambiguation(WSD).

2. Jumi Sarmah and Shikhar Sarma propose a supervised Machine Learning approach- Decision Tree for Word Sense Disambiguation task in Assamese language.[2] A Decision Tree is decision model flowchart like tree structure where each internal node denotes a test, each branch represents result of a test and each leaf holds a sense label. J48 a Java implementation of C4.5 decision tree algorithm is taken for experimentation in their case. A few polysemous words with different real occurrences in Assamese text with manual sense annotation was collected as the training and test dataset. Existing literature reveals that there are various approaches for lexical ambiguity resolution-Knowledge based, Corpus based. In recent years, many WSD systems is being

developed in Indian languages like Hindi, Malayalam, Manipuri, Nepali, Kannada but no such automated system has yet emerged for the Indo-Aryan language- Assamese. Their future work aims to develop a model for the WSD problem which is fast, optimal and efficient in terms of accuracy and scalability. This paper presents a survey report made in this research topic discussing the WSD problem, various approaches along with their algorithms. Moreover it also list out the various NLP applications which would be efficient when disambiguation system is merged.

3. Simone Paolo Ponzetto and **Roberto Navigli** present a methodology to automatically extend WordNet with large amounts of semantic relations from an encyclopedic resource, namely Wikipedia. They show that, when provided with a vast amount of high-quality semantic relations, simple knowledge-lean disambiguation algorithms compete with state-of-the-art supervised WSD systems in a coarse-grained all-words setting and outperform them on gold-standard domain-specific datasets.

4. Manish Sinha, Mahesh Reddy and **Pushpak Bhattacharyya** present an effective method of construction of the Marathi WordNet using the Hindi WordNet both of which are being developed at IIT Bombay. They present Word Sense Disambiguation (WSD) of nouns in Hindi. The system has been evaluated on the Corpora provided by Central Institute of Indian Languages and the results are encouraging.

5. Pradeep Sachdeva ,Surbhi Verma and Sandeep Kumar Singh presents a knowledge based algorithm for disambiguating polysemous words using computational linguistics tool, Word Net. Algorithms in the past have calculated similarity either by finding out the number of common words (intersection) between the glosses (definitions/meanings) of the target and nearby words, or by finding out the exact occurrence of the nearby word's sense in the hierarchy (hypernyms) of the target word's senses. This paper proposes an algorithm which modifies the above two parameters by computing intersection using not only the glosses but also by including the related words. Also the intersection is computed for the entire hierarchy of the target and nearby words. It also incorporates a third parameter 'distance' (between target and nearby words).

6. Eneko Agirre and **German Rigau** presents a method for the resolution of lexical ambiguity of nouns and its automatic evaluation over the Brown Corpus. The method relies on the use oil' the wide-coverage noun taxonomy of WordNet and the notion of conceptual distance among concepts, captured by a Conceptual Density formula developed for this purpose. This fully automatic method requires no hand coding of lexical entries, hand tagging of text nor any kind of training process. The results of the experiments have been automatically

III. Problem Statement

To create a desktop application which can identify and resolve the ambiguity and provide sense of the ambiguous word present in input sentence to user. The task of WSD is to understand the sense of an ambiguous word in a piece of context. It basically assigns the appropriate sense to a word depending on the particular context where it occurs in an automated manner.

IV. Flow Diagram

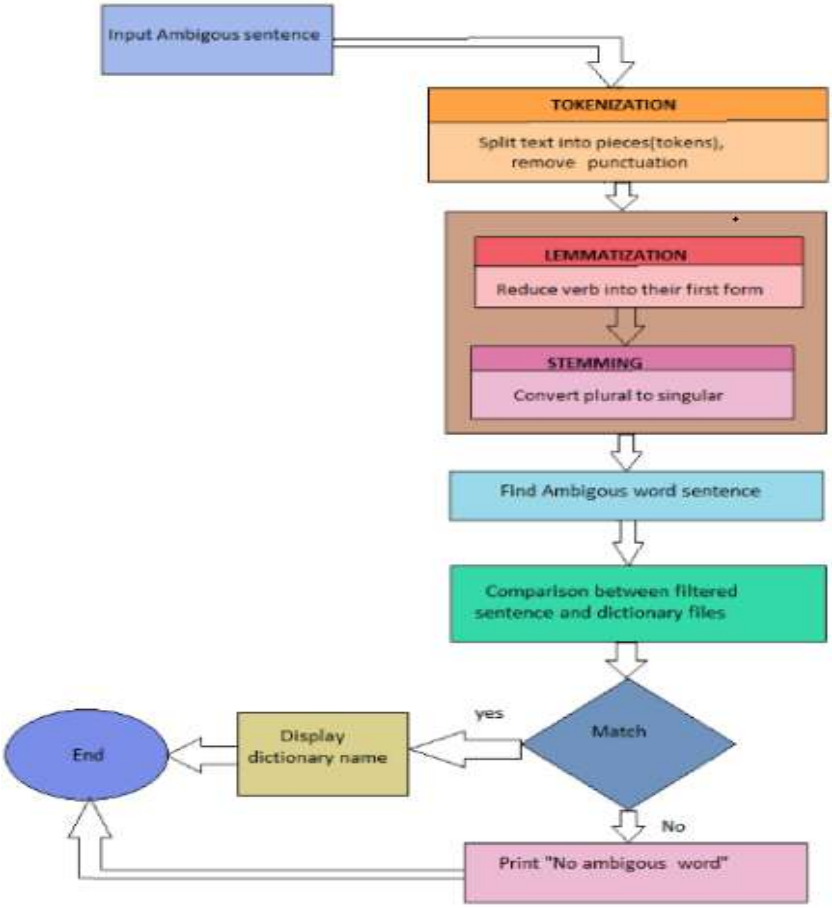


Fig (a): Flow diagram of Word Sense Disambiguation

The application consists of five modules namely Tokenization, Lemmatization and Stemming, Ambiguous word, Dictionary creation and Training. Whenever user finds ambiguity in sentences , user will give those sentences as an input to the application. The application performs tokenization on inputs by removing stop words, special character and articles from sentences. Stopwords, are the high frequency words in a language which do not contribute much to the topic of the sentence. In English, such words include 'a', 'an', 'the', 'of', 'to', etc.. We remove these words and focus on our main subject/topic to solve ambiguity.

Then application performs lemmatization on filtered sentence in which the verbs are converted into their first form. The plural words are converted into their singular form by performing stemming on filtered sentence. Now we have filtered sentence which does not contain plural words.

The third module identifies ambiguous word from filtered sentence. The application has been fed with data set files called dictionary files. The dictionary files are the text files. Dictionary Creation module contain list of dictionary files. The words inside dictionary files are compared with tokens of filtered sentence and if it match then application provide those dictionary file name as an output. At the end user get sense of the ambiguous word without human intervension.

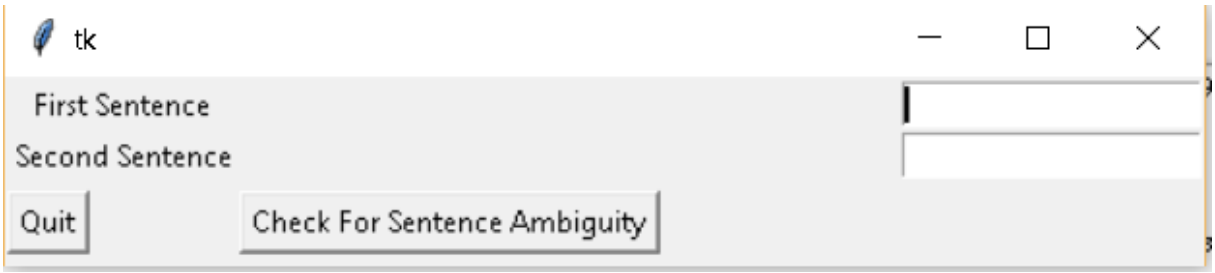
V. Implementation

The domain of this project is Natural Language Processing. An ambiguity is arises in natural language due to different meanings of word present in context. To deal with ambiguity we implemented various steps of Natural Language Processing. The coding for various steps is as shown below:

We have built up the application using various modules using Natural Language toolkit which support Python. Python provides Python Shell which is used to execute a single python command and get the result. The Python Shell performing execution on given input as shown below:

:

The application take two ambiguous sentence as an input.The GUI before taking input is as shown below:



So, now we can click on the Check For Sentence Ambiguity to view the correct sense of ambitious word for respective sentence:

So this is how the implementation has been carried out.

VI. Result Analysis And Discussion

Thus we have developed an application Word Sense Disambiguation(WSD) which disambiguate the word which have multiple sense in a context automatically by identifying and resolving conflict and provide output to user .Every time the user gives the input in terms of sentences, the application resolves the ambiguity and guesses the sense of the particular ambiguous word present in input sentence.

If we give the following sentences as an input

Input 1: Bat is flying

Input 2: Aditya is playing with bat

Then output will be

Output 1: Given sentence is related to mammal bat.

Output 2: Given sentence is related to Cricket bat.

VII. Conclusion And Future Scope

We have implemented desktop application named as Word Sense Disambiguation (WSD) which can identify and resolve the ambiguity and provide sense of the ambiguous word present in input sentence to machine without any human intervention.

We have proposed an approach of Word Sense Disambiguation where our main aim is to identify and resolve ambiguity present in given sentence.

Adding large set of ambiguous words of English language in the application so that machine can resolve ambiguity more efficiently.

Flexibility can be provided to user by supporting application for various languages.

In future, we intend to improve intelligent of system by working on logic based technique as sense dependencies are not always captured in syntactic structure of the sentence.

References

[1]. Ariel Raviv and Shaul Markovitch, Concept-Based Approach to Word- Sense Disambiguation. Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence. Volume 10,22 Jul 2012, PP.807-813.

[2]. Jumi Sarmah and Shikhar Sarma, Decision Tree based Supervised Word Sense Disambiguation for Assamese. International Journal of Computer Applications (0975 – 8887)Volume 141 – No.1, May 2016

[3]. Kalita, P. and Barman. AK, Word Sense Disambiguation: A Survey. International Journal Of Engineering And Computer Science ISSN:2319-7242 Volume 4 Issue 5 May 2015, Page No. 11743-11748V

[4]. Ponzetto, S. P., and Navigli, R. 2010. Knowledge-rich word sense disambiguation rivalling supervised systems. In Proc. of ACL-1, Volume 214,16 Jul 2010.

[5]. Sinha, M., Reddy R.M.K., Bhattacharyya, P., Pandey, P., P., Kashyap, L., www .cfilt .iitb .ac.in/wordnet/webhwn/papers / HindiWSD.pdf, vol-42,P.(2004).
[6]. Devendra Chaplot and Pushpak Bhattacharyya, Unsupervised Word Sense Disambiguation Using Markov Random Field and Dependency Parser, AAAI 2015, Austin Texas, USA, 25-29 January, 2015
[7]. https://towardsdatascience.com/a-simple-word-sense-disambiguation-application-3ca645c56357
[8]. http://www.scholarpedia.org/article/Word_sense_disambiguation.
[9]. Eneko Agirre and German Rigau, "Word Sense Disambiguation using conceptual density," Proceedings of the 16th conference on Computational linguistics - Volume 1, 1996
[10]. Agirre, E.; Lopez de Lacalle, A.; Soroa, A. "Knowledge-based WSD on Specific Domains: Performing better than Generic Supervised WSD". Proc. of l.lCAI, 2009.
[11]. Yarowsky, David. "Unsupervised word sense disambiguation rivaling supervised methods." Proceedings of the annual meeting on Association for Computational Linguistics, 1995.
[12]. Karov, Yael, and Shimon Edelman. "Similarity-based word sense disambiguation." Computational I inguistics 24.1 (1998)