IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 6, June 2016

Ambiguity Resolution in Text Mining Using Probabilistic and Statistical Approach

Parveen[1], Khushboo Chandel[2]

School of Computer Science and Engineering, Bahra University, Shimla Hills, India[1, 2]

Abstract: Ambiguity means inexactness. In case of data mining and natural language processing ambiguity refers to have more than one meaning of the same word. A term 'Polysemy' is also used for such a situation. This paper presents the process of ambiguity removal in data mining (so that the precise meaning of the word should be clear to the computer) and the problems with ambiguity removal. There are different types of ambiguity with a word. We will use probabilistic and statistical approach to remove the ambiguity attached with the word. Word Sense Disambiguation (WSD) is a must condition for human computer interaction otherwise without it the meaning of the word will not be clear to computer as user wants and this will create problems in further interaction between human and computer if user is dealing with a paragraph instead of a sentence. The probabilities for each word can be estimated from a tagged corpus.

Keyword: Ambiguity, POS Tag, Probability, Text mining

1. INTRODUCTION

Sometimes people are facing problems in understanding correct meaning of the sentence. Since, sentence comprises of ambiguous words. In such case, correct meaning is taken by the context of the sentence.[6]It is easy to understand the meaning of a word by human beings but not so easy for the computer, specially if it has ambiguities. The meaning of a word is not clear to the computer if it has multiple meanings or we can say that it has ambiguity. We people can easily understand the word's meaning in which context we are talking about but computer cannot, because the knowledge hub or database or corpus we provide contains the multiple meaning of that word.

Natural language is ambiguous i.e. one phrase often has multiple meanings. Ambiguity is a type of meaning uncertainty giving rise to more than one possible interpretation. Being ambiguous is therefore a semantic attribute of a form whose meaning cannot be resolved according to a rule or process with a finite number of steps. Context, however, plays a role, because something can be ambiguous in one context but not in other. Ambiguity arises when we have more than one parse tree for a sentence, meaning that sentence is ambiguous. In computer science, Natural Language Processing (NLP) struggles a lot with ambiguity.

Making a computer understand about what a sentence, a phrase, a word means is a big challenge. And yet we are fighting against this challenge everyday. For example:-

1) A round table (adjective)
2) A round of cheese (noun)
3) To round out your interests (verb)
4) To work the year round (adverb)

It is a very tough task for the computer to understand that the "round" we used is an adjective, noun, verb or adverb.

The basic definition of ambiguity, as generally used in natural language processing, is .capable of being understood in more than one way. It can be classified into many different types and using various different classification schemes.[2]

• Lexical Ambiguity: is the ambiguity of a single word. A word can be ambiguous with respect to its syntactic class. Eg: book, study.

For eg: The word silver can be used as a noun, an adjective, or a verb.

She bagged two silver medals.

She made a silver speech.

His worries had silvered his hair.

Lexical ambiguity can be resolved by Lexical category disambiguation i.e, parts-of-speech tagging. As many words may belong to more than one lexical category. Part-of-speech tagging is the process of assigning a part-of-speech or lexical category such as a noun, verb, pronoun, preposition, adverb, adjective etc. to each word in a sentence.[4]

• Syntactic Ambiguity : It occurs when a given sequence of words can be given more than one grammatical structure and each having a different meaning . In other words it means how similar are

DOI 10.17148/IJARCCE.2016.56187

two words with respect to their syntactic function or role? It even helps to solve the problem of plagiarisms. The disadvantage of syntactic similarity is that two sentences having the same words in different order can have high syntactic similarity but a completely different meaning.

Example : — SMALL CAR FACTORY — The above sentence can have two meanings . —(small car) factory — OR — (small) car factory —.[5]

Researchers in the text mining community have been trying to apply many techniques or methods such as rule-based, knowledge based, statistical and machine-learning-based approaches. However, the fundamental methods for text mining are natural language processing (NLP) and information extraction (IE) techniques. The former technique focuses on text processing while the latter focuses on extracting information from actual texts. Once extracted, the information can then be stored in databases to be queried, data mined, summarized in a natural language and so on. The use of natural language processing techniques enables text mining tools to get closer to the semantics of a text source. This is important, especially when the text mining tool is expected to discover knowledge from texts.[8]

2. ALGORITHM FOR WORD SENSE DISAMBIGUATION

In this algorithm STEP 1 and STEP 2 can be removed by an already tagged corpus.

Assumptions:-

- k be the number of words in a sentence.
- n be the number of ambiguous words or target words in the sentence.
- P(w/N) is the probability of target word to be noun.
- P(wi) is the probability of the word at index position "i" in the sentence.

STEP 1: Create a knowledge hub or select a corpus.
STEP 2: Apply the POS tagging to the knowledge hub.
STEP 3: Enter the sentence.
STEP 4: Create tokens for the entered sentence.
STEP 5: Apply POS tagging to the entered sentence.
STEP 6: Select the target word say w and return the position of target word in the sentence.
Let that position of target word is "a".
STEP 7: Calculate the probability for the target word to be Noun/Verb.

$$P(N|w)=P(w\cap N)/P(w)$$
$$P(V|w)=P(w\cap V)/P(w)$$

STEP 8: Calculate the probability for the neighboring words or remaining words.

```
        for(i=0;i<k;i++)
        {if(i==a)
          {continue;
          }
          P(wi)=number of words (wi)/total number of words in the corpus;
         }
STEP 9: if(n==1)
        P(w/N)= P(N/w)∏P(wi ) where ∏ varies from i=0 to a-1 and i=a+1 to k-1;
        P(w/V)=P(V/w)∏P(wi ) where ∏ varies from i=0 to a-1 and i=a+1 to k-1;
STEP 10: if(P(w/N)>P(w/V)
         Assign Noun tag to the target word.
         else
         Assign Verb tag to the target word.
```

3. ADVANTAGE OF AMBIGUITY

Many people think that ambiguity creates only problems for the language and the computer but that thing is not true. Ambiguity benefits with one thing that is, it reduces the number of words in the dictionary of a particular language. If a word is ambiguous meaning that there are multiple POS tags that can be assigned to that word during lexical analysis of entered statement or sentence. There does seem to be a consensus in the literature that lexical ambiguity can produce faster lexical decision times, it is not at all clear what type of ambiguity is producing the effect. Is it multiple meanings, or multiple senses that produces the advantage? One way of trying to answer this question is to examine the dictionary entries of the words used in these experiments. As described above, dictionaries make a distinction between words whose meanings are sufficiently unrelated that they are given multiple entries and those that have multiple senses within an entry. This provides a convenient way in which to categories words as being ambiguous between multiple meanings or between multiple senses.[1]Suppose that we have a language which have total 'X' words in the dictionary of that language and out of which 'Y' words are ambiguous. Let 'Z' be the number of unambiguous words which will be Z=X-Y. Let on an average there are 'β' meanings corresponding to an ambiguous word. In such a condition we will need to introduce (β-1)Y extra words so that no ambiguous word present in the dictionary. Therefore the total number of words we will find in that language will be $X=Y+(\beta-1)Y+Z \Rightarrow X= \beta Y+Z$. There are millions of words present in a language and which keep on increasing day by day. For example:- let us suppose there are 900,000 unambiguous and 100,000 ambiguous words in any

DOI 10.17148/IJARCCE.2016.56187

language say English. Corresponding to an ambiguous word say β=3.76(means 3.76 meaning of a single word). Therefore total words that we will need to make the language unambiguous will be X=3.76(100,000)+900,000=1,276,000. This increased our dictionary by 276,000 words. Thus, reduced number of words in a language's dictionary is a strong benefit of ambiguity.

4. PROBLEMS IN AMBIGUITY RESOLUTION AND THEIR SOLUTION

4.1 Word Doesn't Exist: If we have a word(except target word) in the sentence which doesn't exist in the corpus or knowledge hub then in such a case the probability calculated will be zero corresponding to that word. Thus, the probability calculated for the neighboring word in STEP 8 of algorithm will be zero and which also make the probabilities calculated in STEP 9 zero. Therefore we will be not able to make any guess for the word sense regarding word sense disambiguation.

Solution
To overcome this problem we will add a constant named as "probab constant"(whose value is 10-3) in probability of each word(except target word) in STEP 8 of the algorithm. This will remove our problem and we will able to calculate some probability in STEP 9 and therefore we can make decision in STEP 10. Further there is a problem in this, if their exist any corpus such that the probability calculated for a word exceeds 1.0(which is very rare case) after adding probab constant(probability greater than 1.0 not possible for any event). In this situation we will apply a condition in STEP 8 which will assume that if there is any probability exceeds 1.0 then it will be assumed as 1.0.

4.2 Multiple Target Words: If we have a complex problem say any entered sentence with more than one ambiguous words(multiple tags for more than one word) than this algorithm will not able to solve the problem. For example: We have two sentences
1)Innocent peacefully children sleep little.
[('innocent', 'NN'), ('peacefully', 'RB'), ('children', 'NNS'), ('sleep', 'JJ'), ('little', 'JJ'), ('.', '.')]
2)Innocent little children sleep peacefully.
[('innocent', 'JJ'), ('little', 'JJ'), ('children', 'NNS'), ('sleep', 'VBP'), ('peacefully', 'RB'), ('.', '.')]

Solution
To solve this problem we will take a worst case. As the problem is complex, so the solution to this problem will also be complex. Suppose there are "k" words in a sentence. Here the possible worst case is that all the words are ambiguous and here we also suppose that the maximum tags possible to each word is "m". So we have total mk possible cases for which we have to calculate the probability in STEP 7 by making changes which will be(for this we need to call all tags defined in nltk and store them in a variable before this step)

STEP 7: Calculate the probability for the target word for every possible tag.

```
for(i=0;i<k;i++)
{
  for(j=0;j<m;j++)
  {
    P(Tj/wi)=P(wi∩Tj)/P(wi)    /*Calculates the probability of word at index position i with tag at position j */
  }
}
```

```
STEP 8: for(i=0;i<k;i++)
        {
          Search tag with highest probability at index position i and assign that tag to the word at index position i
        }
```

5. CONCLUSION

The above proposed algorithm is very helpful in text mining by reducing the ambiguity of the word or sentence. This algorithm uses probabilistic and statistical approach for the ambiguity removal. The accuracy of the above algorithm depends upon how vast corpus or our knowledge hub is. To interpret the meaning we require a contextual knowledge and world knowledge is required at higher levels. The word sense disambiguation becomes complex if there are multiple ambiguous words in a sentence because the algorithm that will be used for such a case will have to make very much calculations i.e. mk. Therefore the time taken by this algorithm to make the correct decision will be very high. This algorithm will be really helpful in applications like Machine Translation, Information Retrieval, Question Answering etc. by automatic resolution of ambiguities.

REFERENCES

1. Jennifer Rodd,,Gareth Gaskell and William Marslen-Wilson(2013),"The Advantages and Disadvantages of Semantic Ambiguity"
2. Walton Hall and Milton Keynes," Ambiguity Management in Natural Language Generation
3. Dan Roth(1998),"Learning to Resolve Natural Language Ambiguities: A Unified approach". To Appear in AAAI-98

DOI 10.17148/IJARCCE.2016.56187

4. Anjali M K and Babu Anto P," Ambiguities in Natural Language Processing". International Journal of Innovative Research in Computer and Communication Engineering (An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 5, October 2014
5. Shilpi Singh and Dr. Lakshmi P. Saikia(2015)," Ambiguity in Requirement Engineering Documents: Importance, Approaches to Measure and Detect, Challenges and Future Scope". In International Journal of Advanced Research in Computer Science and Software Engineering. Volume 5, Issue 10, October-2015
6. Priti Saktel and Urmila Shrawankar," An Improved Approach for Word Ambiguity Removal".
7. Lidong Bing, Bai Sun, Shan Jiang, Yan Zhang and Wai Lam," Learning Ontology Resolution for Document Representation and its Applications in Text Mining".
8. Shaidah Jusoh and Hejab M. Alfawareh," Techniques, Applications and Challenging Issue in Text Mining".In IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 6, No 2, November 2012.
9. Lars Bungum and Bjorn Gamback," Evolutionary Algorithms in Natural Language Processing".In Norwegian Artificial Intelligence Symposium, Gjøvik, 22 November 2010.
10. Fabrizio Costa, Paolo Frasconi, Vincenzo Lombardo, Patrick Sturt and Giovanni Soda," Ambiguity resolution analysis in incremental parsing of natural language".
11. Daniel Jurafsky and James H. Martin(2015)," Speech and Language Processing"
12. http://www.cs.columbia.edu/~mcollins/hmms spring2013.pdf
13. http://cs.nyu.edu/faculty/davise/ai/ambiguity.html
14. http://www.nltk.org/
15. http://www.illc.uva.nl/LaCo/CLAS/ull14/papers/chap12.pdf
16. http://stp.lingfil.uu.se/~nivre/docs/statnlp.pdf

17.http://www.cse.unsw.edu.au/~billw/cs9414/notes/nlp/ambiguity/ambiguity-2012.html

18.http://cs.williams.edu/~andrea/cs108/Lectures/NLP/NLP.html

DOI 10.17148/IJARCCE.2016.56187