Word sense disambiguation algorithms by artificial intelligence

Mahdi Alinaghizadeh Ardestani[1], *Zhila Mohammadi*2*

1. Faculty Member, electical Engineering, Technical and Vocational University (TVU), Tehran, Iran
2.P.H.D, Department of Basic Sciences, , Technical and Vocational University (TVU), Tehran, Iran

Abstract

The existence of words with the same spelling and different meanings in a sentence causes problems in understanding the meaning and machine translation of that sentence, and in special circumstances, in intelligent systems and semantic web applications and search engines, it is necessary to disambiguate these words. Word sense disambiguation is a core research problem in computational linguistics, which was recognized at the beginning of the scientific interest in machine translation and artificial intelligent .In the field of computational linguistics, the problem is generally called word sense disambiguation (WSD), and is defined as the problem of computationally determining which “sense” of a word is activated by the use of the word in a particular context. WSD is essentially a task of classification: word senses are the classes, the context provides the evidence, and each occurrence of a word is assigned to one or more of its possible classes based on the evidence. Words are assumed to have a finite and discrete set of senses from a dictionary, a lexical knowledge base, or an ontology (in the latter, senses correspond to concepts that a word lexicalizes). Application-specific inventories can also be used. For instance, in a machine translation (MT) setting, one can treat word translations as word senses, an approach that is becoming increasingly feasible because of the availability of large multi-lingual parallel corpora that can serve as training data. The fixed inventory of traditional WSD reduces the complexity of the problem and making it tractable.In this article, an innovative method is presented in order to resolve the semantic ambiguity of ambiguous words in a sentence. This method is based on using the knowledge available in the dictionary and using fuzzy logic, the ambiguous and non-ambiguous word or words of the sentence are identified and categorized by the dictionary. Then, in the next step, disambiguation of ambiguous words is done by having a suitable dictionary. Fuzzy logic is also used to increase accuracy in choosing the correct meaning of ambiguous word . The effectiveness of the proposed method has been evaluated using data collected from authoritative sources.

Keywords: natural language processing, disambiguation, artificial intelligence, fuzzy logic.

Introduction

Word meaning is in principle infinitely variable and context sensitive. It does not divide up easily into distinct sub-meanings or senses. Lexicographers frequently discover in corpus data loose and overlapping word meanings, and standard or conventional meanings extended, modulated, and exploited in a bewildering variety of ways [5]. In lexical semantics, this phenomenon is often addressed in theories that model sense extension and semantic vagueness, but such theories are at a very early stage in explaining the complexities of word meaning [7]. WSD has obvious relationships to other fields such as lexical semantics, whose main endeavour is to define, analyze, and ultimately understand the relationships between “word”, “meaning”, and “context”. But even though word meaning is at the heart of the problem, WSD has never really found a home in lexical semantics. It could be that lexical semantics has always been more concerned with representational issues [4] and models of word meaning and polysemy so far too complex for WSD . And so, the obvious procedural or computational nature of WSD paired with its early invocation in the context of machine translation [2] has allied it more closely with language technology and thus computational linguistics. In fact, WSD has more in common with modern lexicography, with its intuitive premise that word uses group into coherent semantic units and its empirical corpus-based approaches, than with lexical semantics [11]. Writing has long been used to document science. The process of learning, discovering and extracting sciences is always done by studying and reviewing written texts. Nowadays, due to the high volume of data production, the need for faster information processing has become very important. Due to the large volume of data, the use of automatic and machine knowledge discovery and extraction methods has been considered. One of the important processes in text data is the translation of a text from one language to another. In all the official languages of the world, there are words that have different meanings despite having the same written structure. These words are known as ambiguous words. The operation of finding and assigning the correct meaning of an ambiguous word is called disambiguation. Resolving semantic ambiguity depends on the text. Based on the process of learning throughout life and acquiring knowledge, humans acquire the ability to recognize different meanings of an ambiguous word in a text. Machine disambiguation methods are divided into three categories: 1- Knowledge-based methods 2- Methods based on sentence structure and body 3- Creative and combined methods [8].

A Brief History of WSD Research

The studies and research done in word ambiguity have a long history in linguistics and natural language processing. In general, machine disambiguation methods are divided into two general categories and a combined category. In knowledge-based methods, by having a dictionary as a source of knowledge, a decision is made about the meaning of a word [12]. In knowledge-based methods, the knowledge source (dictionary) can be updated by the learning capability for the algorithm. The learning operation can take place in the entire execution of the algorithm, which is known as the unsupervised method, and there is no separate step for learning at the beginning of

the algorithm. In the learning step, based on a data set whose words have already been disambiguated by the experts and the algorithm starts learning. In this way, methods with an supervised are called [7]. In corpus-based methods, the position of ambiguous words relative to other words in the sentence is measured and decisions are made based on the position of the words. In these methods, learning can be used as an auxiliary parameter in the algorithm [8]. In general, the success of the learning step is more in the observer methods. The disadvantages of this type of learning can be mentioned such as increasing the time complexity of the algorithm and the limited training set. The number of ambiguous words in some languages, such as Persian, is high, which reduces the efficiency of this method [10]. WSD was first formulated as a distinct computational task during the early days of machine translation Kaplan (1950) and Reifler (1955)) recognized the basic statistical character of the problem in proposing that "statistical semantic studies should be undertaken, as a necessary primary step. The 1950s then saw much work in estimating the degree of ambiguity in texts and bilingual dictionaries, and applying simple statistical models. Zipf (1949) published his "Law of Meaning"4 that accounts for the skewed distribution of words by number of senses, that is, that more frequent words have more senses than less frequent words in a power-law relation-ship; the relationship has been confirmed for the British National Corpus [9] . Kaplan (1950) determined that two words of context on either side of an ambiguous word was equivalent to a whole sentence of context in resolving power. Some early work set the stage for methods still pursued today. Masterman(1957) represent the different sense of a word, and then chose the heading whose contained words were most prominent in the context. Madhu and Lytle (1965) calculated sense frequencies of words in different domains – observing early on that domain constrains sense – and then applied Bayes formula to choose the most probable sense given a context. WSD was resurrected in the 1970s within artificial intelligence (AI) research on full natural language understanding. In this spirit, Wilks (1975) developed "preference semantics", one of the first systems to explicitly account for WSD. The system used selectional restrictions and a frame-based lexical semantics to find a consistent set of word senses for the words in a sentence. In 1986, Lesk introduced an algorithm in which the meanings of words are determined by the dictionary. Each individual meaning is compared to the definition of that word in the dictionary in terms of other similar words. The meaning that has the most similarity (the most overlap) with other words is chosen as the correct meaning [11].

research method

The framework and general outline of the proposed method is shown in Figure (1). In the first step, raw text is given as input to the system. In the proposed method, it is assumed that the raw text is a text that has not been edited and contains letters, numbers and symbols. Also, the text is without spelling mistakes. This text is written in Persian language.

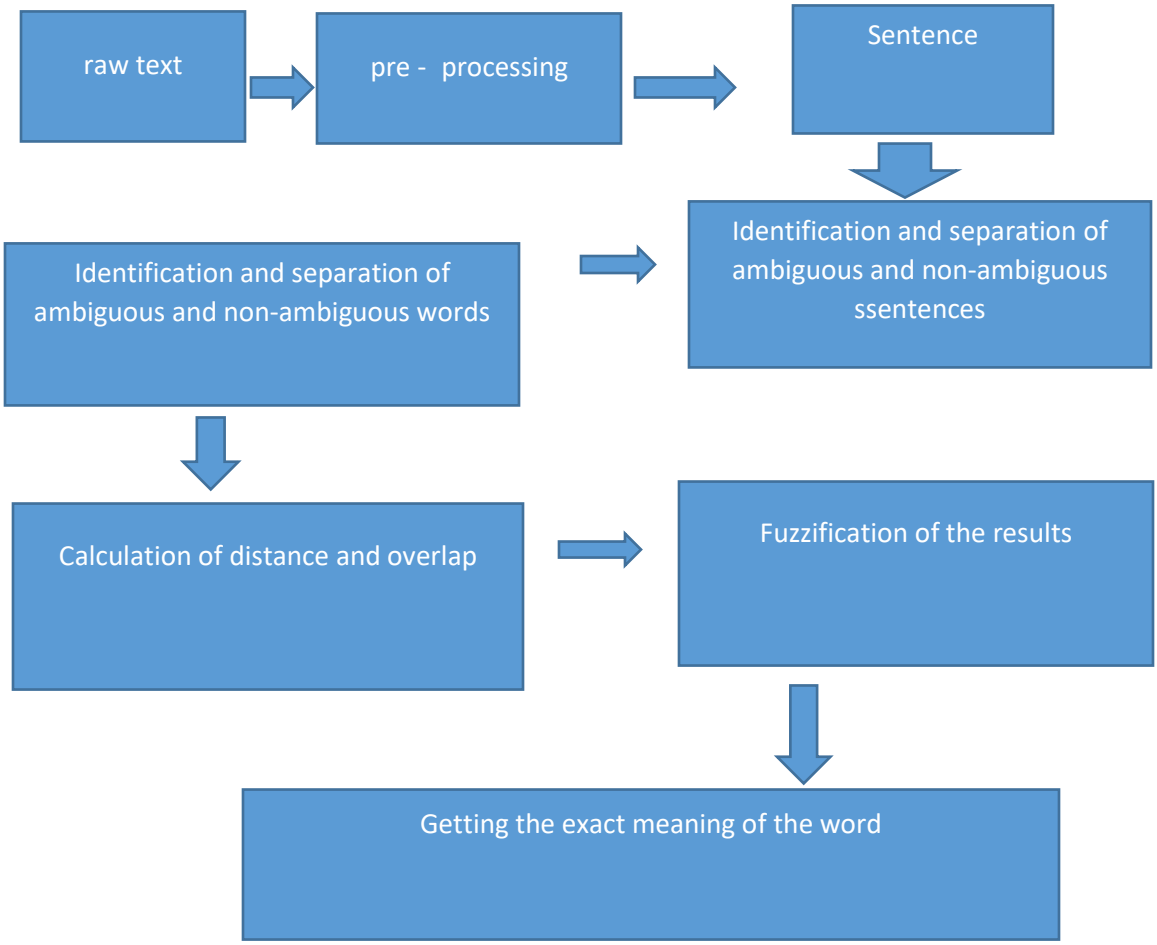

Figure (1): The general framework of the proposed method of removing semantic ambiguity

In the pre-processing step, the input text is refined. Text preprocessing is not only an essential step to prepare the corpus for modeling but also a key area that directly affects the natural language processing (NLP) application results. Preprocessing is the most important step in text mining, natural language processing and information retrieval. Text preprocessing refers to a series of techniques used to clean, transform and prepare raw textual data into a format that is suitable for

NLP or ML tasks. The goal of text preprocessing is to enhance the quality and usability of the text data for subsequent analysis or modeling. In this step, the text is refined by prepared list of words. Prepositions and conjunctions, despite being frequent in the sentence, have little semantic value, and for this reason, they are removed in the pre-processing stage in natural language processing applications. The list includes prepositions, conjunctions, pronouns, auxiliary verbs, conjunctions, numbers and non-Persian words. After the pre-processing, the resulting text contains only the words that are not in the desired list, performing the pre-processing step reduces the calculation load and increases the speed and accuracy of the subsequent processing steps.

In the next step, the refined text is phrased. For this purpose, a tool called sentence separator is used. Separators are one of the most important natural language processing tools. This tool has the ability to recognize sentences and extract them from the text according to the characters that separate the sentence (in Persian, such as question mark). At the end of this step, a list of sentences is extracted. The step of identification and separation is one of the most important steps of the algorithm to distinguish ambiguous from non-ambiguous sentences and to separate them. In this section, ambiguous sentences are identified by the list of ambiguous words and the list of sentences extracted in the sentence phase then they are labeled. In the proposed method, it is assumed that the sentence contains one or more ambiguous words. The method is that if there is a word from the list of ambiguous words in the sentences, that sentence is labeled as an ambiguous sentence and is added to the list of ambiguous sentences. At the end of this step, a list of ambiguous sentences is extracted. In the next step, the distance between the non-ambiguous words and the ambiguous word of the sentence and the amount of overlap are calculated. In the proposed method, the distance of the number of non-ambiguous words is used to measure the distance. In an ambiguous sentence, each non-ambiguous word has a specific distance from the ambiguous word. The distance between the non-ambiguous word and the corresponding ambiguous word is inversely proportional to its effectiveness in the disambiguation process. For example, the ambiguous sentence "a lion is an animal that lives in the forest" after preprocessing becomes the sentence "an animal , a lion, forest , life" and the distance between the unambiguous words "animal", "forest" and "life" from the ambiguous word "lion" is respectively 1, 2 and 3 and its effectiveness is 3, 2 and 1. A dictionary is used to calculate the overlap criterion. In the dictionary, different and complete definitions are provided for each ambiguous word. In an ambiguous sentence, the overlap of each of the non-ambiguous words for each of the meanings of the ambiguous word in this dictionary is calculated. by the dictionary and having non-ambiguous words. The calculation result is obtained by the overlapping function. The overlap function has three outputs:

Output zero when the non-ambiguous word is not in the dictionary. Output 1, if exactly the word other than to exists in the desired dictionary and a value between zero and one when there is a similarity between the ambiguous words in the dictionary and the sentence. This similarity gives additional help to disambiguate the ambiguous word. In many sentences, there is non-ambiguous word in the plural form or a state where the structure of the word has changed a little (for example, in a sentence there is unambiguous word forests, but in the dictionary there is forest). In this case,

if the only goal is to find the same word in the sentence and the dictionary, a non-ambiguous word that may play an important role in solving the ambiguity is omitted. For example, the ambiguous sentence "a lion is an animal that lives in the forest" has the following three different meanings in the dictionary. The lion is a large cat of the genus Panthera, native to Africa and India. It has a muscular, broad-chested body; a short, rounded head; round ears; and a hairy tuft at the end of its tail. It is sexually dimorphic; adult male lions are larger than females . Milk is a white and nutritious liquid with a sweet taste that is secreted from the breasts of female mammals. Valve: The compound word "Shiralat" is widely used in Persian today. The output of the overlapping function for the non-ambiguous word "forest" for the three different meanings of the mentioned lion is as follows:

$$f_{o}(w_{۱} = \text{'lion'}) = ۱, f_{o}(w_{۱} = \text{'milk'}) = 0, f_{o}(w_{۱} = \text{'valve'}) = 0$$

In the fuzzification of the results, the fuzzy approach is used to solve the ambiguity problem. At first, the distance and overlap criteria in the previous step are converted into fuzzy variables. The distance of a non-ambiguous word to the ambiguous word relative to the total number of non-ambiguous words in the sentence is converted into a continuous value between zero and one by formula (2).

$$f(dis) = \frac{\text{distance}w_i}{\text{n} - \text{word}}$$

In this formula, $\text{distance}w_i$ is the distance between the non-ambiguous word and the ambiguous word in position i. n_word is the number of non-ambiguous words in the sentence .

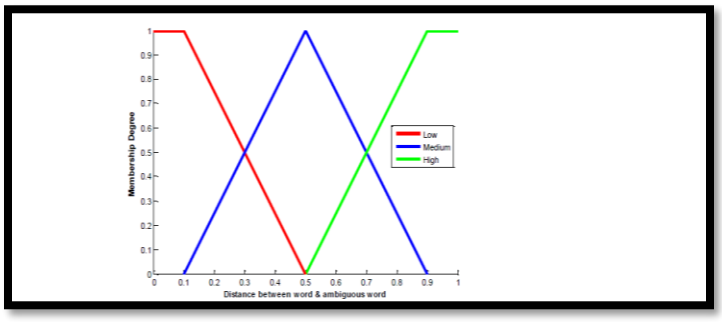

Figure (2): The diagram of the functions describing the fuzzy distance between unambiguous words and ambiguous words

In the next step, the overlap of non-ambiguous words in the dictionary is checked. Non-zero output values and an overlap function are mapped with a step function to the fuzzy descriptor with "low, medium and high" values. Also, for values of one, the output from the overlap function is also mapped to the "more" descriptor value. If the similarity ratio between the non-ambiguous word in the sentence and the dictionary is between 4.0 and 5.0, it is considered as low ; the similarity ratio is between 5.0 and 7.0, as moderate similarity and more than this value as high similarity.

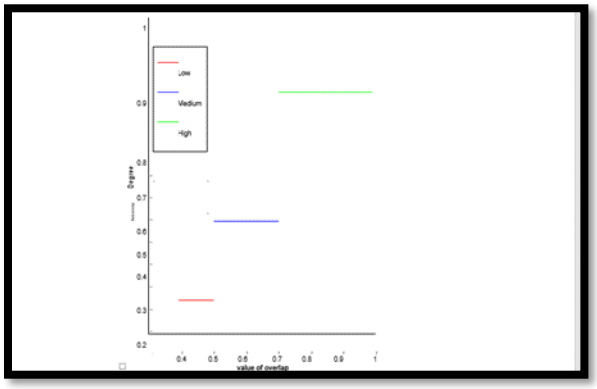

Figure (3): Diagram of functions describing fuzzy overlap of unambiguous words in the dictionary.

After this step, the calculated distance value from formula (2) is converted into a fuzzy descriptor with "low, medium and high" values by a triangular function and two trapezoidal functions. In the last step (the step of extracting the exact meaning of the word), if-then rules are used to select the meaning of the word. Table (1) shows the fuzzy if-then rules used in the algorithm. Mamdani implication is used to evaluate if-then rules. For each non-ambiguous word and its relationship with the meaning of the ambiguous word, a weight is obtained based on table (1). The smaller the distance between ambiguous and non-ambiguous words and the higher the percentage of similarity of the ambiguous word found in the dictionary, the more weight is given to its meaning. At the end of the evaluation step for each meaning, the weight of non-ambiguous words are added together and whichever meaning gets more weight is selected as the output of this step.

Weight meaning Word	unambiguous word overlap in dictionary	Unambiguous word-to-word distance

	word	**ambiguous**
1	low	low
2	average	low
3	high	low
1	low	average
2	average	average
3	high	average
1	low	high
2	average	high
3	high	high

data analysis

At this stage, the proposed method implemented by MATLAB software is compared with Bayesian method [2] and fuzzy method using neural network [3]. In the simulation, a list of 10 ambiguous Persian words according to table (2) is used. Table (3) compares the accuracy percentage of choosing the correct meaning of the proposed method with the other two methods. According to table (3), the proposed method has higher accuracy in disambiguation of ambiguous words.

Ambiguous words	Number of meanings
lion	3
garlic	2
livestock	2
species	2
barley	2
shapes	2
light	2
seal	2
breath	2
salty	2

Conclusion

Disambiguation is considered a main step in machine translation and text analysis. Several methods have been proposed to solve this problem. In this article, a fuzzy method is used to disambiguate ambiguous words. The results of the experiments show the superiority of the proposed method in removing the ambiguity of words compared to other presented methods.

Ambiguous words	suggested method	Bayesian method	Total text method
lion	83%	78%	74%
garlic	85%	79%	76%
livestock	89%	89%	80%
species	85%	81%	86%
barley	86%	87%	78%
shapes	85%	83%	71%
light	86%	79%	76%

seal	84%	79%	79%
breath	86%	81%	82%
salty	81%	78%	73%

Table 2- The list of ambiguous words used in the simulation

References

[1] Ide, Nancy. Veron, Jean. (1998) “Introduction to the Special Issue on Word Sense Disambiguation: The State of the Art, Journal Computational Linguistics, Vol. 24, Pp. 2-40.
[2] Makki, Raheleh, omayounpour, Mohammad Mehdi. (2008) “Word Sense Disambiguation of Farsi Homographs Using Thesaurus and corpus”, 6th International Conference, GoTAL, Pp. 315–323.
[3] Lesk, M. (1986) "Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone", Proceedings of the 5th annual international conference on Systems documentation, Pp. 24-26, New York, NY, USA. ACM.
[4] Hazrati Fard, M. Fakhrahmad, S.M. Sadreddini, M.H. (2014) “Word Sense Disambiguation based on Gloss Expansion” 6th Conference on Information and Knowledge Technology.
[5] Ranjan Pal, Alok. Saha, Diganta. (2015) “WORD SENSE DISAMBIGUATION: A SURVEY”, International Journal of Control Theory and Computer Modeling (IJCTCM) Vol.5, No.3.
[6] Agirre, Eneko. Edmonds, Philip. (2007) “Word Sense Disambiguation; Algorithms and Applications”, Springer, Vol. 33.
[7] Navigli, R. (2009) “Word Sense Disambiguation: a Survey”, ACM Computing Surveys, Vol. 41, No.2, Pp. 1-9.
[8] Cucerzan, R.S., C. Schafer, and D. Yarowsky, (2002) “Combining classifiers for word sense disambiguation”, Natural Language Engineering, Vol. 8, No. 4, Cambridge University Press, Pp. 327-341.
[9] Fakhrahmad, S.M. Rezapour, A.R. Zolghadri Jahromi, M. Sadreddini, M.H. (2011) "A new word sense disambiguation system based on deduction," Proceedings of the World Congress on Engineering, Vol. 2, Pp. 1276-1281.
[10] Rasekh, A.H. Sadreddini, M.H. (2013) "Word Sense Disambiguation Algorithms Based on the Context, Structure and Meaning," International Journal of Signal and Data Processing, Vol. 2, Pp. 40-47.

[11] McRoy, Susan. 1992. Using multiple knowledge sources for word sense discrimination. Computational Linguistics, 19(1):1–30.

[12]Turing, Alan M. 1950. Computing machinery and intelligence. Mind, 59:433–460.