Malaysian Journal of Computing, 3 (2): 154–161, 2018

eISSN: 2600-8238 online

WORD SENSE DISAMBIGUATION USING FUZZY SEMANTIC-BASED STRING SIMILARITY MODEL

Amir Abd-Rashid[1], Shuzlina Abdul-Rahman[2], Nor Nadiah Yusof[3] and Azlinah Mohamed[4]
[2]Research Initiative Group of Intelligent Systems,
[1,2,3,4]Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA,
40450 Shah Alam, Selangor, Malaysia
[1]are_mer10@yahoo.com, [2]shuzlina@tmsk.uitm.edu.my,
[3]nornadiah@gmail.com, [4]azlinah@tmsk.uitm.edu.my

ABSTRACT

Sentences are the language of human communication. This communication medium is so fluid that words and meaning can have many interpretations by readers. Besides, a document that consists of thousands of sentences would be tough for the reader to understand the content. In this case, computer power is required to analyse the gigantic batch size of the text. However, there are several arguments that actively discuss regarding the output generated by a computer toward the meaning of the passage in terms of accuracy. One of the reasons for this issue is the existing of the ambiguous word with multiple meanings in a sentence. The passage might be incorrectly translated due to wrong sense selection during the early phase of sentence translation. Translating sentence in this paper means either the sentence has a negative or positive meaning. Thus, this research discusses on how to disambiguate the term in a sentence by referring to the Wordnet repository by proposing the use of fuzzy semantic-based similarity model. The proposed model promising to return a good result for detecting the similarity of two sentences that has been proven in the past research. At the end of this paper, preliminary result which shows the flow of how the proposed framework working is discussed.

Keywords*: Disambiguity, Fuzzy, SentiWordNet, WordNet*

1. Introduction

According to Oxford dictionary, the definition of a sentence is a series of words begin with a capital letter that consists of various types of words such as noun, verb, adjective, adverb and conjunction for delivering information to the reader. A sentence without any ambiguous words can be easily understood. However, a sentence that consists of ambiguous words cannot be directly understand without considering other words in the same sentence. Ambiguous refers to a situation where a word has a multiple senses (meaning) and multiple words might have the same senses (Liu et al., 2007; Yusof et al., 2015). There are four main types of sentences that are commonly known, which are simple sentence, compound sentence, complex sentence and compound-complex sentence. Since a sentence is in a natural language form, thus it is not easily 'understood' by a computer. There is a massive amount of studies that has been done in this research area, and a lot more are still ongoing.

A technique proposed in this research is the fuzzy semantic-based string similarity for word sense disambiguation (WSD). WSD is a very important process to be done before further process while fuzzy approach suitable to automate range selection process (Asraf et al., 2017; Yahaya et al., 2017). This fuzzy method also brought human intelligent to support decision making (Ismail & Syaiful 2015). This technique is used to support sense selecting in WordNet repository.

WordNet is a very useful library to be used in completing this study. There is a set of cognitive synonym known as synsets that represent noun, verb, adjective and adverb for each English word. Another repository that is very resourceful is SentiWordNet. SentiWordNet is an extended repository for WordNet that provides polarity scores for each English word. The

polarity score is kept for each word that will represent the meaning of the word where it is inclined towards a positive or negative meaning.

The repository is widely adopted since it provides a broad coverage lexicon for English words (Wang et al., 2017).

However, SentiWordNet only provides lexicons in a semi-automatic manner. It requires some methods for doing a selecting task to get the right sense for a word. There is a huge volume of words that carry more than one sense. For example, the word "like" can be defined as ~~a~~ "similar" or "find enjoyable or agreeable". Both definitions are obviously different and will influence the whole sentence meaning. This aspect needs to be considered while deriving sense from the SentiWordNet repository. The problem is known as word sense ambiguities. Thus, to overcome this issue, a process of word sense disambiguation (WSD) is required. WSD algorithms goal to resolving word ambiguity (Basile et al., 2014; Razak et. al., 2018).

There are various formulae and algorithms that has been proposed in previous study to derive prior polarity of the word from SentiWordNet repository. However, the performance varies depending on the adopted variant. Gatti et al. (2016) stated that the existing method still has a deficiency which still can be enhanced to complete the missing piece. They also proved that the existing formulae are outperformed on other data sets. These findings has shown that there is still a lot of research that needs to be conducted to resolve this issue.

This paper aims to demonstrate the proposed approach for WSD using fuzzy semantic-based approach. Section 2 of this paper will briefly describe the approach that has been proposed and comparison in terms of the differences between the proposed and existing methods. The following section, which is section 3 describes about WordNet and SentiWordNet. In section 4, the results are discussed and finally, section 5 concludes the paper.

2. Proposed Approach

This research focuses on Word Sense Disambiguation (WSD) problem in which the formulae and algorithms in handling these problems are highlighted. Overview of the existing approach has been elaborated in detail by Li et al. (2017).

One commonly known algorithm in handling the WSD problem is Lesk Algorithm. This algorithm was introduced as early as in 1986 by a researcher named Lesk. Since then, the algorithm has been enhanced and upgraded (Bordes et al., 2010). One of the approaches that was implemented in Lesk Algorithm is by comparing the similarity of the sense definition. In this approach, the definition of each word in the sentence are compared by cross-checking if there is any of definition of each word is the same with other words' definition. Further elaboration and discussion about this approach are available in Klapaftis & Manandar (2005). The proposed idea in this paper, which to compute the similarity between words is derived from the idea of Lesk algorithm. However, Lesk algorithm makes use of word definition. On the hand, the proposed method will use the sense definition for each word to compare the similarity with input sentences which contains the ambiguous word.

The term 'ambiguous' in this research refers to the multiple meanings of words that have been described in WordNet. The nature on how WordNet keeps all the list of synset is the main reason that contributes to the WSD process. More details about the nature of WordNet will be described in the next section. This is also mentioned by Bird et al. (2009) in the elaboration section of this topic. For example, the word "fight" in WordNet has five senses in total for noun tag.

Table 1: Sense of word "*Fight*"

Sense	+ve	-ve	Example Sentence
battle.n.01	0.0	0.0	"a hostile meeting of opposing military forces in the course of a war"
fight.n.02	0.0	0.0	"the act of fighting; any contest or struggle"
competitiveness.n.01	0.0	0.125	"an aggressive willingness to compete"
fight.n.04	0.125	0.25	"an intense verbal dispute"
fight.n.05	0.0	0.0	"a boxing or wrestling match"

Table 1 shows the breakdown of the list of senses for the word "fight". The example of the input sentence: "one of his teammates was disqualified from continuing playing since he triggered the fight toward his opponent". The input sentence will be compared with the example of the sentence for each sense by using the Fuzzy Semantic-Based String Similarity. Compared to Lesk Algorithm, the model is quite different since Lesk Algorithm compares each sense definition with the neighbouring word sense definition. Lesk Algorithm aims to get the most similar sense definition for each word in the same sentence. The sense with the most similar definition will be selected to contribute to the meaning of the sentence. However, in this proposed approach, the input sentence will be compared to the example of the sentence for each word.

The Fuzzy Semantic-Based String Similarity is commonly implemented to solve plagiarism detection problems. Alzahrani & Salim (2010) has conducted research that implementing Fuzzy Semantic-Based String Similarity to detect plagiarism and has adapted in nearly 80 continuation research in related topic. The similarity of a pair of a sentence can be determined from two types of similarities, which are semantic and order. In this research, only the semantic vector will be considered in deriving the sentence similarity. The semantic vector of two sentences is obtained by using a unique term from both sentences along with their synonyms and similarities from WordNet. Since WordNet has provided a function to get the similarities of two words, it will be adopted to this research to get the word similarity. The formula of similarity for each sentence is derived by taking the average similarity of each word in a sentence, see Equation (1).

$$S = \frac{\sum w_n}{n} \tag{1}$$

The formulae (Eq. 1) represent the overall similarity for each sentence. S represents similarity value for a sentence, which is derived from the summation of w_n, where each word represents similarity and n represents the number of words to compare. Similarity values will always be in the range of 0 to 1 which are equivalent to $\mathbf{0 < S < 1}$. The assumption for this research is that sense with a sentence that has the highest similarity value with the input sentence is the most significant sense to be selected that contributes the meaning for the input sentence.

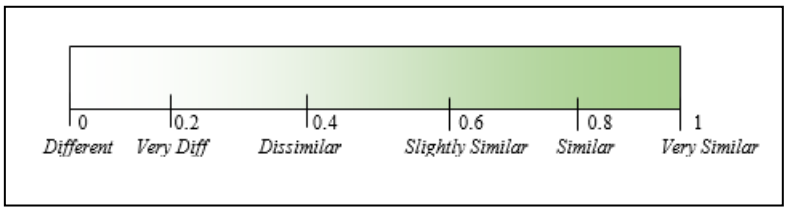

Figure 1: Fuzzy Value Indicator

Figure 1 indicates the rule break for a fuzzy value to identify the similarity of the input and sense sentences. A fuzzy value is divided into 6 scales for determining the similarities of the sentences. For each similarity, the sentence will be listed and compared to the most similar for sense selection. Word with scores between 0 - 0.4 will be disqualified for further process for contributing to sense selection. This range contains the fuzzy value of "Very Different", "Different" and "Dissimilar" which mean the source word is completely different from the target word. Words with a score of 0.4 - 1 will be considered as contributing to the word sense selection. This range has the fuzzy values of "Slightly Similar", "Similar" and "Very Similar", which mean the source word has similarity value to the targeted word. The same rule break for fuzzy value was implemented by Straccia (2016). However, not all fuzzy indicators were used in this research. The following equation is the mapping on similarity acceptance of each word.

$$s = \begin{cases} \text{reject if } s(score) < 0.4 \\ \qquad\quad \text{where } 0 < s < 1 \\ \text{accept} \quad otherwise \end{cases} \tag{2}$$

The mapping represents on rule break of filtering sense that has a similarity or is non-similar. Rejected sense will not be stored wherever on the list while the acceptance sense will only be stored for further processes. More details of the process for sense selection will be described in the next paragraph.

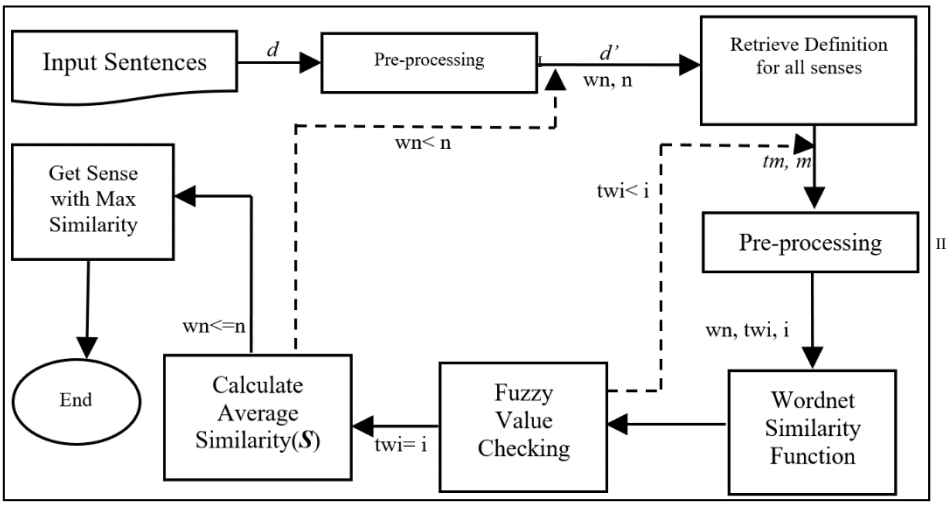

Figure 2: System Framework

Figure 2 represents the whole framework of WSD process by implementing the Fuzzy Semantic-Based String Similarity. The input sentence is labelled as 'd', where it needs to be pre-processed before it proceeds to chunk every useful word that will give meaning to the sentence. This process is also known as stop word elimination process, where 'd' is an input sentence after eliminating all the stop words. This step is important since stop words do not bring any meaning to the sentence (Jamaludin et al., 2013). Even though, if stop words are not eliminated from the sentence before proceeding to the next step, the meaning of the sentence remains the same as the removed stop word version. It is a very significant practice to remove stop word in this step to avoid extra loop in the next process which can consume more computing resources as a result. Technically, most of the standard process flow will start from the pre-processing phase.

The second step in this model is to retrieve definition sentences for all senses for each word in the input sentence. Variable '***wn***' represents current word count in iteration while '***n***'

is the number of all words to be processed in the sentence. Words that are stored in a list are brought to the next process in the loop. It is significantly an efficient way to update any value that associates them to the current word in sequence. This behaviour will make the further process simple and easy to handle any debugging operation. Since it is a list, thus '***wn***' initiate location is 0. Traditional looping behaviour is applied to retrieve the definition sentences for each sense. As mentioned earlier, each word has more than one senses.

However, some senses might contain empty example of definition sentences whilst some others might contain more than one example of definition sentences. This is one of the challenges that might affect the accuracy of the final selection. For each sense's definition sentences are held by variable '***tm***' and they are the same as previous, '***m***' is the number of total sense's definition sentence to be processed. The sense of each word from the example sentence is also require subjected to the pre-processing phase to eliminate all the stop words that are found in the sentence. Again, the sense of each word in the example sentence will be stored in a list where where '***twi***' and '***i***' are list sizes. This entire process requires recursive loop to process each word. For each input sentence, word list and sense's definition sentence word list will be processed to acquire the semantic similarities between each pair of words.

3. Result and Discussion

This section presents the preliminary results of the proposed approach on a small size of documents. In this experiment, a set of documents from movie review data is used. Let's say, the example of a sentence is given as, *"he is playing as a team fight"*. For this sentence, 3 words are kept after pre-processing i.e. playing, team, fight; thus, for the first iteration, similarity measurements will run through for the word *"playing"*. This word consists of 3 sets of senses in WordNet repository which are:

P1: *"the act of playing a musical instrument"*
P2: *"the action of taking part in a game or sport or other recreation"*
P3: *"the performance of a part or role in a drama"*

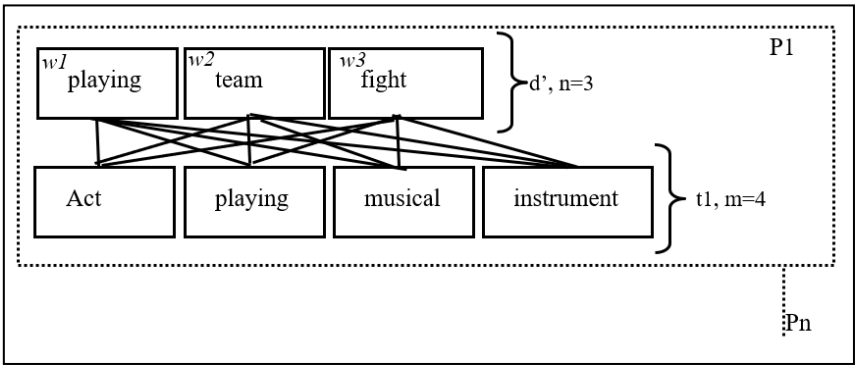

Figure 3. Word similarity process

Figure 3 represents the list of useful words in the input sentence being a process in the first iteration with '***w1***' sense definition. Results for this iteration will be run through to the fuzzy semantic-based method to select the sense with sufficient similarity for contributing the meaning for sentences. Sample output for figure 3 is represented in Table 2.

Table 2 shows the tabulations of the similarity results for the first iteration. The overall results indicate that only three pairs of the words return value greater than 0.4. The three pairs are: Playing-Playing, Playing-Fight, Musical-Fight. In this case, the word 'Playing' has 3 senses, thus 3 iterations need to be done i.e. first sense return word "Act", "Playing", "Musical" and "Instrument". The similarity for each word vs input sentence word is shown in Table 2.

Table 2 Example of Similarity Value

	Playing	Team	Fight
Act	0.2106	0.2667	0.2666
Playing	1	0.2222	0.4760
Musical	0.381	0.2666	0.5333
Instrument	0.1	0.1333	0.1333

Table 2 above shows the tabulation of the similarity result for the first iteration. The overall result indicates that only three pairs of the word return value greater than 0.4.
Thus, after the sentence similarity calculation is applied, the similarity value is 0.6698 which the calculation is as follows: (1 + 0.4760 + 0.5333)/3 = 0.6698. This is the similarity value for this sense. However, this value needs to be compared with the other two iterations for this sense. The highest similarity value will be selected as the most suitable sense for the word.

After the sense selection, the score of sense will be stored in the list for all word in the same sentence for each input sentences. The selected sense will contribute meaning to the sentence either the sentence has a positive or negative meaning. The dataset that will be used in this research has already defined for each document if the whole document is negative or positive. Thus, it will consider as a benchmark to verify the accuracy of this WSD model

4. WordNet and SentiWordNet

WordNet is a large repository lexical relation of English relation. WordNet is widely adapted to various researches in natural language processing (NLP) area. This repository provides lexical for almost every word with part of speech of noun and verb. Each lexical has set of lemma-Pos that sharing the identical meaning that's known as 'synset' (Bird et al., 2009). SentiWordNet is a sub-library in WordNet that contains a set of scores for each synset in WordNet. For each synset has pre-assigned positive and negative score that has a range of 0 – 1. Each word might have more than one synset. This is also known as an ambiguous listing. However, each synset has sorted corresponding to the most frequently used sense is label lowest lemma#Pos#sense-number (Wang et al., 2017). As mentioned in the previous section, the reason of WSD process is to select the most suitable sense that contributes the real meaning for the sentences.

SentiWordNet is a sub-library of WordNet that contains sets of scores for each synset in WordNet. Each synset has pre-assigned positive and negative scores that range from 0 – 1. Each word might have more than one synsets. This condition is also known as an ambiguous listing. However, each synset is sorted corresponding to the most frequently used sense, which is labelled as the lowest lemma#Pos#sense-number (Wang et al., 2017). As mentioned in the previous section, the reason for the WSD process is to select the most suitable sense that contributes to the actual meaning of the sentences.

WordNet is also prebuilt with various libraries with a function that relates to NLP. One of the functions adopted by this research is a function to get the similarities between two words. The function mentioned is labelled as *wup_similarity* in WordNet. This label stands for "Wu-Palmer Similarity" where it will return a score denoting how similar two-word senses are, based on the depths of the two senses in the taxonomy and that of their Least Common Subsume. There is also another method used to get the similarity of two words, which is the Pedersen's Perl implementation. However, not all results produced by *wup_similarity* is agreed by consistent with Pedersen's Perl implementation. These phenomena in this research area are normal, where not all implementation is agreed by another researcher for some reason or another. However, the *wup_similarity* function is still legitimate since it is still being used by various implementations that adapt WordNet as a repository of the English Language lexical source.

5. Conclusion

This research has proposed the use of the fuzzy semantic-based similarity model to resolve the word sense disambiguation problems. WSD is needed to ensure that the right senses are selected before the whole sentence can be determined whether it is a positive or negative sentence.

Without proper implementation of WSD, each word in the sentence might be assigned with the wrong sense, which may not be even related to what the actual sentence means. Eventually, the sentence may be evaluated on a different degree of the score from what it supposed to be. For future work, large volume of documents can be exercised and compared with the Lesk algorithm.

Acknowledgement

The main author would like to express their sincere appreciation to all those involved in the preparation of this research paper. Thank you for all the contributions in giving good ideas and providing guidance to ensure the completion of this research paper. Hopefully, the collaboration in the successful completion of this study will contribute to the accomplishment of other studies.

References

Alzahrani, S., & Salim, N. (2010). Fuzzy semantic-based string similarity for extrinsic plagiarism detection. *Braschler and Harman, 1176*, 1-8.

Asraf, M. H., Dalila, N. K., Faiz, A. Z., Aminah, S. N., & Nooritawati, M. T. (2017). A fuzzy inference system for diagnosing oil palm nutritional deficiency symtoms. *ARPN Journal of Engineering and Applied Science, 12(10), 3244-3250.*

Basile, P., Caputo, A., & Semeraro, G. (2014). An enhanced lesk word sense disambiguation algorithm through a distributional semantic model. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers* (pp. 1591-1600).

Jamaludin, N. A., Annamalai, M., Jamil, N., & Bakar, Z. A. (2013, December). A model for keyword Profile Creation using extracted keywords and terminological ontology. In *e-Learning, e-Management and e-Services (IC3e), 2013 IEEE Conference on* (pp. 136-141). IEEE.

Bird, S., Klein, E., & Loper, E. (2009). *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.".

Bordes, A., Usunier, N., Weston, J., & Collobert, R. (2010) Learning to Disambiguate Natural Language Using World Knowledge.

Gatti, L., Guerini, M., & Turchi, M. (2016). SentiWords: Deriving a high precision and high coverage lexicon for sentiment analysis. *IEEE Transactions on Affective Computing*, *7*(4), 409-421

Ismail, M. and L. Syaiful (2015). Affective assessment in learning using fuzzy logic. e-Learning, e-Management and e-Services (IC3e), 2015 IEEE Conference on, IEEE.

Klapaftis, I. P. and S. Manandhar (2005). Google & wordnet based word sense disambiguation. Proceedings of the 22nd International Conference on Machine Learning Workshop on Learning and Extending Ontologies by using Machine Learning Methods.

Li, X., et al. (2017). "Prior Polarity Dictionary Derived from SentiWordNet based on Random Forest Algorithm." 2nd International Conference on Automation, Mechanical Control and Computational Engineering (AMCCE 2017) **118**: 818-824.

Liu, Y., Scheuermann, P., Li, X., & Zhu, X. (2007, May). Using wordnet to disambiguate word senses for text classification. In *international conference on computational science* (pp. 781-789). Springer, Berlin, Heidelberg.

Razak, Z. I., Abdul-Rahman, S., Mutalib, S., and Abdul Hamid, N. H. (2018). Web Mining In Classifying Youth Emotions. Malaysian Journal of Computing. 3(1), 1-11.

Straccia, U. (2016). *Foundations of fuzzy logic and semantic web languages*. Chapman and Hall/CRC.

Wang, X., Tang, X., Qu, W., & Gu, M. (2017, October). Word sense disambiguation by semantic inference. In *Behavioral, Economic, Socio-cultural Computing (BESC), 2017 International Conference on* (pp. 1-6). IEEE.

Yahaya, F., Rahman, N. A., & Bakar, Z. A. (2017). Resolving Malay Word Sense Disambiguation Utilizing Cross-Language Learning Sources Approach. *Advanced Science Letters*, *23*(11), 11320-11324.

Yusof, N. N., Mohamed, A., & Abdul-Rahman, S. (2015, September). Reviewing classification approaches in sentiment analysis. In *International conference on soft computing in data science* (pp. 43-53). Springer, Singapore.