Developing Word Sense Disambiguation Corpuses using Word2vec and Wu Palmer for Disambiguation

Fadli Husein Wattiheluw
Department of Informatics
Institut Teknologi Sepuluh Nopember, Surabaya , Indonesia
fadliwattiheluw1994@gmail.com

Riyanarto Sarno
Department of Informatics
Institut Teknologi Sepuluh Nopember, Surabaya , Indonesia
riyanarto@if.its.ac.id

Abstract*— In computational linguistics, meaning disambiguation is an open problem of natural language processing in the form of the process of identifying the meaning of the word polysemy used in a sentence. Resolving this problem, among others, has an impact on search engine relevance, anaphoric solving, coherence or cohesion, and inference or conclusion. Therefore, a study is needed that studies to find the meaning of a correct word on a topic. So that it affects the topics discussed in a sentence to find the true meaning. In this study, we focused on finding the meaning of words in a corpus-based sentence using word2vec and wu palmer. The word2vec algorithm is used to construct word vectors contained in sentences and wu palmer as an addition to new words that are not contained in the corpus, by assessing hypernym, meronym, and hyponym between words in sentences. The experimental results show that by adding a new word using wu palmer on corpus it can increase the precision value of 0.8232 in an introduction to a sentence contained in a topic, compared to not using the addition of a new word.

Keywords—Word sense disambiguation, hyponym, meronym, hypernym, wu palmer, word2vec.

I. INTRODUCTION

Also known as sentiment analysis is opinion mining refers to process determining opinions or emotions expressed in a text about the subject. Although sentiment analysis is a field of research recently, which was introduced in 2001 [1] This raised a lot of interest and many applications to know the sentiments on the opinion of users, for example, in the product reviews, news, twitter, and blog [2].

In determining better sentiments in user comments, the word disambiguation plays a role in sentiment to determine the meaning of a word in a sentence. Word Sense Word Sense Disambiguation (WSD) is an ability to identify the meaning of words in computing [3]. There are many polysemic words that have different meanings for each topic. Sometimes it is not easy for a computer to identify the meaning of some polysemic words on a particular domain so that it requires a model that is used to overcome the problem of polysemic words.

For example, there were two sentences:

(1) I and my friend stayed in room 201 for 2 nights.

(2) My friend was treated in room 201 for 2 nights.

In the sentence, there is the word “room” which has a different meaning in hotel and hospital domain. The word “room” in the first sentence describes room for traveling, while. The word ' room ' in the second sentence explains the care of the sick. Computers will have difficulty determining the meaning of a sentence well if there is a polysemic word. Therefore, it has become a major problem in several studies of natural language processing. Techniques of word sense disambiguation is an automatic way to determine meaning word a context in opinion. Generally, WSD is identifying which sense of a word is used in a sentence when the word has multiple meanings.

The last few years there is a research about WSD, such as that done by Bagus Setya [4]. This research is concerned with the improvement of model graph-based approach WSD, weight graph extracted by using some measure of similarity (i.e.: Leacock & Chodorow, Wu Palmer, Resnik, Lin And Jiang & Conrath). to improve the model by increasing the lesk algorithm with adapted lesk [5]. As for other studies conducted by Amita, Devendra, and Sonakshi [6] distinguish words in wordnet uses fuzzy semantic relations. Where the early stages of defining words in the lexical categories. Furthermore, Identify the words that stand out in the context of the collective. After narrowing the categories, understanding of the word is found using a modified lesk algorithm. As for other studies conducted by Su and Thanda [7] focused on both supervised and knowledge-based approaches. With the new coefficient-based WSD algorithm proposed to overcome match vocabulary problems. External knowledge resources corpus and wordnet are used as a repository of sense by linking with the new WSD algorithm to consider additional semantics for WSD.

Other studies related to the development of corpus conducted by Fika, Riyanarto, and Chastine [8]. They built the corpus to detect emotions in documents using a model called Corpus-Based of Emotion (CBE). CBE developed from Affective Norms for English Words (ANEW) and Wordnet Affect Emotion (WNA) with the term similarity and distance approaches the size of the node. in addition to the research conducted by Fika, automatically adding new words that are not found on the CBE corpus using Latent Dirichlet Allocation (LDA). As for other studies to classify emotions in the music domain using lyrics and audio as a feature [9]. The lyric feature is extracted from text data and audio features extracted from audio signal data. In the classification of emotions, the emotional corpus is needed for the extraction of lyrical features. Corpus-Based Emotion (CBE) managed to increase the value of F-Measure for classification emotions in text documents. Music documents have an unstructured format compared to article text documents. So that it requires a good preprocessing and conversion process before the classification process. The best test results for the classification of musical emotions are the application of the Random Forest method for lyrics and audio features.

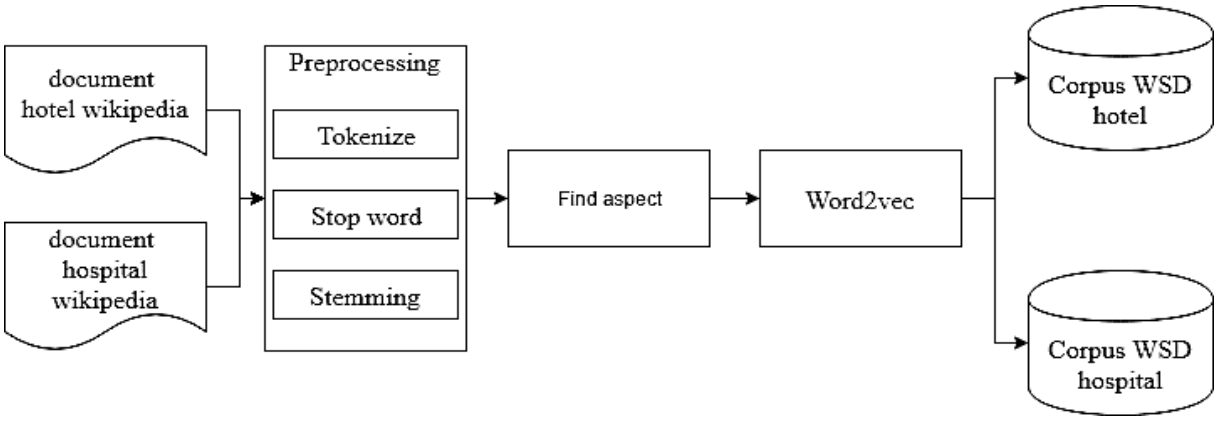

Fig. 1. Step develop corpus WSD

Research conducted by Endang [10]. They proposed a new lexical database called B-BabelNet. The model proposed to improve analysis of semantic business process model. They try to map Wikipedia pages to WordNet database but only focus on words related to the business domain. In addition, to enrich vocabulary in the business domain. They also use terms in specific online business dictionaries. The results in the disambiguation process using B-BabelNet show an increase in the accuracy of meaning disambiguation, especially in certain matters related to business and industrial domains.

Problems often encountered in natural language processing are in determining the meaning of a polysemic word which means it will be different in a topic or domain. So it requires a sense of disambiguation to distinguish corpus-based words for each domain. Therefore, this study proposes that the WSD corpus is built using the Word2vec algorithm for the initial stage of the WSD corpus formation and adds a new word using Wu Palmer. To expand the WSD corpus built for a particular domain. There are several stages to build a WSD corpus for a particular domain. First, every training document taken from Wikipedia will be preprocessed. Then each word will be carried out by the training process using the word2vec algorithm to build vector terms. To be able to find out the topic on the testing document, the similarity value will be calculated based on the word vector on each topic. to compare documents with certain topics using several important words as aspects that explain the topic. To find the important word, we use the term frequency technique to find important words with the frequency of occurrences on the topic. The words that are not in the WSD corpus are built on certain domains. To overcome words that are not in the corpus, we use the Wu Palmer algorithm to find the similarity of the meaning of the word. To add new words to the WSD corpus by calculating the value of the similarity between words based on hypernym, hyponym, and meronym values.

This paper is organized as follows: Part II describes the stages of creating a corpus and determines the meaning of words on the topic. Section III describes the analysis and evaluation of the results of the proposed method. And the last part IV describes the conclusions from the results of the experiment.

II. METHODS

In developing the WSD corpus, the method proposed is divided into two main parts: (A) At the stage to build a WSD corpus that uses Wikipedia as training data. Training data will be preprocessed, search for important words and build vectors for each word on the topic using word2vec. (B) At this stage, we automatically add new words that are not in the WSD corpus. The process of adding new words using the Wu Palmer algorithm by considering the value of similarity between hypernym, hyponym, and meronym. In this study will focus on hotel and hospital domains in overcoming WSD.

A. Develop corpus WSD

The training document will be carried out in the preprocessing process (ie tokenize, stemming, and stop word) to eliminate unnecessary words in building WSD corpus. After the preprocessing process, the next is to determine important words based on the number of events on the topic of hotels and hospitals using the term frequency technique. After determining important words for each topic, we build vectors for each word using the word2vec algorithm. Word vectors contained in the WSD corpus are used to distinguish the same word for different topics.

Figure 1 illustrates the initial development stage for the WSD corpus in handling polycemic words. Where training data is taken manually from Wikipedia with hotel and hospital keywords. The number of datasets taken about hotels and hospitals with a total of 134 documents used as training data to develop the WSD corpus. After collecting data for training, the next stage will be carried out. Data processing starts with tokenizing, stemming, and deleting irrelevant words in the WSD corpus. The next stage, the preprocessing results document will be carried out in the process of finding important words in each document that is used as an aspect of the topic of hotel and hospital. After finding the word as an aspect, then from the training data will be carried out in the process of vector formation using the skip-gram model for each word in hotel and hospital documents. The last stage, every word that has been trained will have vector values for hospital documents and hotels

that will be stored as word sense. As for the explanation of each process:

1) Preprocessing

Training document data from Wikipedia will be broken down into a term called tokenize. Furthermore, unimportant terms contained in the tokenize results to be deleted are called stop words such as a the, in, for, etc. Data has been done the stop word process then undertaken the process of stemming, whereby every word will be returned to basic Word.

For example, there is a document "A hotel is an establishment that provides paid lodging on a short-term basis". Document conduct done process tokenize, where a sentence would be broken into words. Tokenize results "A", "hotel", "is", "an", "establishment", "that", "provides", "paid", "lodging", "on", "a", "short", "term", "base". After done tokenize on a document. The next step, namely the stop word process to eliminate unimportant words such as on, a, is, that. The results of the stop word as follows: "hotel", "establishment", "provides", "paid", "lodging", "short", "term", "base". And the last stage of preprocessing that is stemming. Whereby, every word quickly became basic words into: "hotel", "establish", "provide", "paid", “lodge”, “short”, "term", "stale". This preprocessing step we use natural language toolkit python library in helping our work.

2) Find word aspect in the topic

After preprocessing data on hotel and hospital training from Wikipedia. The next step, we look for candidate words that are used as aspects presented about hotels and hospitals. In defining word used as an aspect to present hotel or hospital. we use the Term Frequency (TF) to find candidate words that are used as aspects. Where words have more frequent occurrences for a topic in hotel and hospital. To search for important words about hotel and hospital topics, we use the following equation:

$$tf(w_t, d) = \frac{\sum_{t=1}^{p} w_{t,t}}{N} \quad (1)$$

Where $tf(w_t, d)$ is the number of frequencies of words that appear on a document, N is a number of words in document d and $w_{t,t}$ is the number of occurrences of words w_t in document d.

TABEL 1. TERM FREQUENCY

Topic	Term			
	establish	provide	paid	Room
Hotel	1	3	4	6

On table 1 displays term frequently appears on the topic hotel. Where the term "space" has a number of occurrence frequencies greater than other words in the hotel topic. The value of "room" term frequency using equation (1) has a value of 0.428 which is used as an aspect presented about the hotel. The results of searching for words as aspects that we use on the topic of hospitals and hotels can be seen in table 2.

3) Word2vec

After doing the stages to look for word candidates as aspects that present topic of hotel or hospital. next is to create a vector for each word that presents words about the topic of hotels and hospitals. This stage we use script-gram model algorithms on word2vec [11] to find the vector representation of word *w(t)* to predict surrounding words in a sentence. As shown in figure 2.

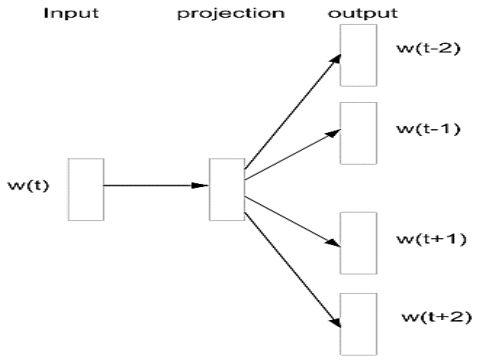

Fig. 2. The Skip-gram model architecture.

To find a word representation is useful for predicting the surrounding words in a sentence or document. given a sequence of word training $w_1, w_2, w_3, \ldots, w_t$. In maximizing the average log probability using the equation [11] :

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\le j\le c, j\neq 0} \text{Log}\, P(w_{t+j}|w_t) \quad (2)$$

Where c the context is the measurement of training (that could be a function of the word center w_t). Formulation of Skip-gram basis defines $P(w_{t+j}|w_t)$ using softmax function:

$$p(wO|wI) = \frac{\exp(v'_{wo}\, v^T_{wI})}{\sum_{w=1}^{W} \exp(v'_w\, v^T_{wI})} \quad (3)$$

where v_w and v'_w are “input” and “output” vector representations of w. w is a number of words in the vocabulary. The results of vector formation for each word for hotel and hospital topics, we use the gram-skip model on word2vec as shown in table 2, where every word has a vector representation. Utilizing the Word2vec algorithm, we use the Gensim library in python to help our work build vectors for each word on the topic of hotel and hospital.

4) Polysemic word sense

After creating a vector for each word presented on the topic of hotels and hospitals using word2vec. Then each word will be saved into the bin file that is used to load a collection of word2vec result vector words. With a vector which is used as a corpus word sense disambiguation to the topic of hotel and hospital. As for example words and vector are stored on each topic of hotel and hospital that can be shown in table 2.

TABEL 2. VECTOR WORD ASPECT REPRESENTATION HOTEL AND HOSPITAL

Topic	Aspect Term	Vector Word
Hospital	Room	[0.2520761 -0.08401345 -0.12540267 ... 0.10789914 0.0204985 0.36947712]
	laboratory	[0.27022526 -0.10543724 -0.13813923 ... 0.0681444 -0.01654486 0.42020062]
	Service	[0.46512705 -0.12488033 -0.18997538 ... 0.13502474 0.02251283 0.5396639]
	Sleep	[0.49783045 -0.20110206 -0.29111138 ... 0.15726066 0.0426247 0.65003633]
Hotel	Room	[-0.01425708 0.01081498 0.02258818 ... 0.02870315 0.00616242 0.00703874]
	Area	[0.02792124 0.00325425 0.02134224 ... 0.03123865 0.00904845 0.00219979]
	Service	[0.0408654 0.02672396 0.03365852 ... 0.01243731 0.00323341 -0.02636981]
	Facility	[0.0261229 0.01244247 0.01922303 ... -0.02984453 -0.02988675 0.03401406]

Each word has a different vector for hospital and hotel topics. Every word has a different vector w_t although the same word on different topics. For example, the word "room" has a vector [0.2520761-0.08401345-0.12540267 ... 0.10789914 0.0204985 0.36947712] in the hospital. But the vector of the word "room" in the hotel domain will be different from the vector of the word "room" in the hospital domain. To find out the topic in the sentence, we use the following equation:

$$P(d|z) = \frac{\sum_{i=1}^{n} SIM_{(wt,z)}}{N} \quad (4)$$

Where N is the number of words in the document, $P(d|z)$ is the probabilistic document on the topic, dan $SIM_{(wt,z)}$ is the value of the similarity of the word w_t with the topic.

B. Automatic expand polysemic WSD

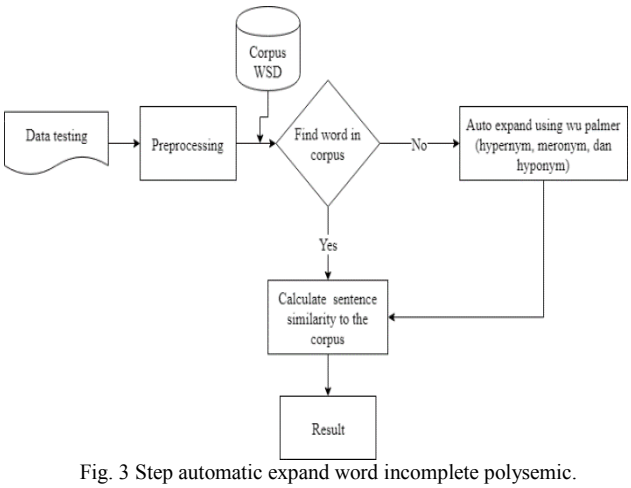

Fig. 3 Step automatic expand word incomplete polysemic.

5) Auto expand corpus

At this stage, we process the addition of new words if the search term is not found in the WSD corpus. Where a new word will do the search process the biggest similarity values using the algorithm Wu Palmer [12] with the equation:

$$WUPw_t = \frac{2 \times N}{N1 + N2} \quad (5)$$

Where N is the parent of the first word of a line $N1$ with the second word $N2$, $N1$ is the number of the line into the first word, dan $N2$ is the number of the line into the second word.

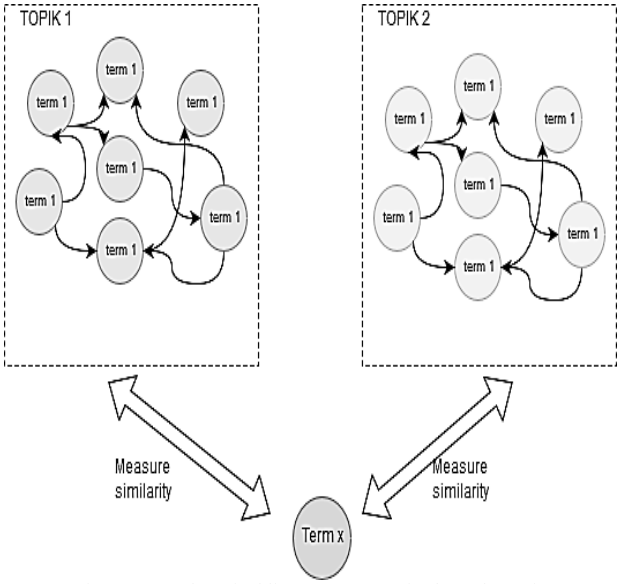

Fig. 4 Examples of adding new term x in the polysemic

In figure 4, displays the term x which is not included in topic 1 or in topic 2. To learn this term included in one topic or both, we use the Wu Palmer algorithm. First, we use some words on the topic to be used as a comparison of the similarity with the term x. Next, we calculate the value of greatest similarity between term x with the term on the topic. If the term x values greater similarity to the term in topic 1. then term x is part of topic 1 and uses vector term on topic 1.

6) Calculate sentence similarity

Finding similarity of words on the topic, we use the cosine similarity algorithm [13] in calculating the value of each word vector as an aspect using the equation:

$$SIM_{(wt,z)} = \frac{\sum_{i=1}^{n} wt_i \times z}{\sqrt{\sum_{i=1}^{n} wt_i^2} \times \sqrt{z^2}} \quad (6)$$

Where $SIM_{(wt,z)}$ is the value of the similarity between the words w_t to the topic z, z is the vector of the topic as a reference classification document, wt_i is a vector on the word i.

III. RESULT

In this study, we took data testing from Twitter using Twepy crawler in python applications. Data obtained from crawler results on twitter as many as 539 data based on hotel and hospital topics. Each test data will be classified on the topic using cosine equations based on word vectors that have been made with word2vec. In this study to evaluate the proposed method, we use a confusion matrix by looking for precision, recall, and F-measure.

TABEL 3. COMPARISON RESULT OF THE PROPOSED METHOD

Method	Precision	Recall	F-Measure
Word2vec	0.7915	0.9	0.8423
Word2vec + Wu Palmer	0.8232	0.9481	0.8812

The results are shown in table 2 in making the WSD corpus using word2vec have less accuracy. We don't pay attention to new words that are not contained in corpus WSD. The precision value obtained is 0.7915 using word2vec without the addition of a new word. It will be different if the WSD corpus is built with attention to new words that are not in the corpus. To add a new word in the corpus WSD, we use Wu Palmer algorithm to pay attention to meaning between words. Corpus WSD built using word2vec and wu palmer can increase precision values by 0.8232. So that it can improve in classifying the meaning of words on a better topic. When compared without using the addition of a new word on the WSD corpus automatically.

IV. CONCLUSION

From the results of experiments conducted in making corpus for disambiguation by considering the new term that will be added to the corpus. A new term to WSD corpus is added using Word2vec and Wu Palmer. Where word2vec algorithm is used as an initial stage of creating a vector for each term on the topic; then, Wu palmer algorithm is used to add a new term to expand the corpus. The combination of Word2vec and Wu Palmer algorithm can achieve higher precision and recall than those achieved by using only Word2vec to build the corpus WSD.

REFERENCES

[1] S. R. Das and M. Y. Chen, "Yahoo! for Amazon: Sentiment Extraction from Small Talk on the Web," *Manage. Sci.*, vol. 53, no. 9, pp. 1375–1388, 2007.

[2] K. Dave, K. Dave, S. Lawrence, S. Lawrence, D. M. Pennock, and D. M. Pennock, "Mining the peanut gallery: Opinion extraction and semantic classification of product reviews," *Proc. 12th Int. Conf. World Wide Web*, pp. 519–528, 2003.

[3] E. W. Pamungkas and D. G. P. Putri, "An experimental study of lexicon-based sentiment analysis on Bahasa Indonesia," *Proc. - 2016 6th Int. Annu. Eng. Semin. Ina. 2016*, pp. 28–31, 2017.

[4] B. S. Rintyarna and R. Sarno, "Adapted weighted graph for Word Sense Disambiguation," *2016 4th Int. Conf. Inf. Commun. Technol. ICoICT 2016*, vol. 4, no. c, 2016.

[5] S. Banerjee and T. Pedersen, "An Adapted Lesk Algorithm for Word Sense Disambiguation Using WordNet," *Comput. Linguist. Intell. Text Process.*, vol. 2276, pp. 136–145, 2002.

[6] A. Jain, D. K. Tayal, and S. Vij, "Word sense disambiguation using fuzzy semantic relations," *Proc. 10th INDIACom; 2016 3rd Int. Conf. Comput. Sustain. Glob. Dev. INDIACom 2016*, pp. 2197–2202, 2016.

[7] S. M. Tyar and T. Win, "Jaccard coefficient-based word sense disambiguation using hybrid knowledge resources," *2015 7th Int. Conf. Inf. Technol. Electr. Eng.*, pp. 147–151, 2015.

[8] F. H. Rachman, R. Sarno, and C. Fatichah, "CBE: Corpus-based of emotion for emotion detection in text document," *Proc. - 2016 3rd Int. Conf. Inf. Technol. Comput. Electr. Eng. ICITACEE 2016*, pp. 331–335, 2017.

[9] F. Hastarita Rachman, R. Sarno, and C. Fatichah, "Music Emotion Classification based on Lyrics-Audio using Corpus based Emotion," *Int. J. Electr. Comput. Eng.*, vol. 8, no. 3, pp. 1720–1730, 2018.

[10] E. W. Pamungkas, R. Sarno, and A. Munif, "B-BabelNet: Business-Specific Lexical Database for Improving Semantic Analysis of Business Process Models," *TELKOMNIKA (Telecommunication Comput. Electron. Control.*, vol. 15, no. 1, p. 407, 2017.

[11] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "502 Distributed Representations of Words and Phrases and Their Compositionality," pp. 1–9.

[12] K. Manjula Shenoy, K. C. Shet, and U. D. Acharya, "A New Similarity Measure for Taxonomy Based on Edge Counting," *Int. J. Web Semant. Technol.*, vol. 3, no. 4, pp. 23–30, 2012.

[13] G. Sidorov, A. Gelbukh, H. Gómez-Adorno, and D. Pinto, "Soft similarity and soft cosine measure: Similarity of features in vector space model," *Comput. y Sist.*, vol. 18, no. 3, pp. 491–504, 2014.