# Word Sense Disambiguation in Bahasa Indonesia Using SVM

Edi Faisal
Department of Computer Science
Dian Nuswantoro University
Semarang, Indonesia
faisal@dinus.ac.id

Farza Nurifan
Department of Informatics
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
farzanurifan@gmail.com

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

***Abstract*—Currently, the use of Indonesian language on the internet is growing rapidly and many of the existing sentences contain ambiguous words. In Natural Language Processing the problem to find out the meaning of an ambiguous word is called Word Sense Disambiguation. Word sense disambiguation is a problem about how we know the meaning of an ambiguous word in a given sentence. Many uses if we can solve word sense disambiguation problems such as can be used for text classification, text clustering and for machine translation. In this paper, we propose the use of SVM algorithm with TF-IDF as the feature extraction method and Wikipedia as the training data to solve the WSD problem of Indonesian language. The results of our proposed method reach an accuracy level of 0.877.**

***Keywords— Word Sense Disambiguation; SVM; TF-IDF; Wikipedia***

## I. INTRODUCTION

Internet usage in Indonesia is currently growing very fast. Lots of information in the form of sentences spread through social media by the Indonesian people. From that sentence, we can analyze various things one of them is the analysis of sentiment[1]. But if there is an ambiguous word in a sentence then we will be difficult to analyze it. The field that handles the problem is word sense disambiguation (WSD) where we make the computer understand what the ambiguous word really means from a sentence. For example, in the sentence:

1. "Malam ini bulan purnama terlihat indah."
2. "Malam hari di bulan Juni terlihat indah."

From the above sentence, the word "bulan" in Bahasa Indonesia has multiple meaning. In the first sentence "bulan" means "Earth satellites" and in the second sentence "bulan" means "Time".

Implementation of word sense ambiguation is widely used in information retrieval, lexicography, knowledge mining / acquisition, and semantic interpretation, and in current emerging studies such as bioinformatics and the Semantic Web. Word sense disambiguation also becomes the main solution in overcoming the ambiguity of words or sentences such as text classification [2] and text clustering [3].

Many researchers have proposed various methods to solve word sense disambiguation in various languages outside English such as building a corpus for supervised word sense disambiguation for the Arabic language [4] with Arabic wordnet mapping method and Wikipedia to select Wikipedia article that matches with wordnet. Furthermore, the cross-lingual method is used to measure the similarity between the feature of Wikipedia article with wordnet. In addition, WSD is also used for classification in Turkish [5]. Classification is done with 6 (six) machine learning method that is C4.5, Random forest, Rocchio, Naïve Bayes, KNN, linear and multilayer perceptron. WSD is also used to build corpus word sense disambiguation in Hindi [6], [7]. Sense annotated Hindi corpus developed from Hindi wordnet which has been available in porta Technology Development for Indian Languages (TDIL). Word sense disambiguation for Assam languages [8] using machine learning Naive Bayes Classifier. While Rini Wongso [9] using machine learning for news classification in the Indonesian language regardless of word ambiguation. The TF-IDF and SVD algorithms are used for feature selection and classification methods using Multinomial Naïve Bayes, Multivariate Bernoulli Naïve Bayes, and Support Vector Machine. Benchmarking of this method is used to find the best classification method.

Navigli [10] makes 3 main approaches in word sense disambiguation: knowledge-based, unsupervised and supervised approaches. According to Navigli [10] supervised approached is the best approach among the three approaches above, provided that the training data used is data with good quality. With reference to this study using data training from the Wikipedia site with the consideration he is a repository of the largest knowledge encyclopedias available for free in 282 languages, has high editorial quality [11], has many uses like to improve semantic analysis [12], and already has a category that can be used as a label. There are many another approach for word sense disambiguation. For example, using adapted weighted graph [13] or using degree and dependency structure of association between sentences [14] for word sense disambiguation.

In this paper, we propose the use of SVM algorithm with TF-IDF as the feature extraction method and Wikipedia as the training data to solve the WSD problem of Indonesian language. We use SVM because in our test it has been shown that this is the best classification that has the highest accuracy. First, we search Wikipedia articles to use it as the training data. The Wikipedia articles selection is based on the category label

from the articles itself. The articles are then preprocessed to make the words variation to be fewer. After that we use TF-IDF to convert the articles into classifiable features and then we determine the meaning of an ambiguous word in a sentence with Support Vector Machines classifier.

This paper is structured as follow; Section II is the Methodology; Section III is the Result; Section IV is the Conclusion

## II. METHODOLOGY

In this paper, we divide our method process into five parts; first is Wikipedia articles selection; second is training and testing data; third is Preprocessing; fourth is Machine Learning using SVM; fifth is performance measure.

### *A. Wikipedia articles selection*

The selection for Wikipedia article is done by first searching for a predefined ambiguous word, in this case, "motor" which has the ambiguous meaning of motorcycle (vehicle) and propulsion (electric). It is also the word "bulan" that has ambiguous meanings of time and earth satellites.

Next look for Wikipedia articles that match the ambiguous words above by looking at the categories in the Wikipedia article which will be used as a label of the sentence that contains the word ambiguous. From the articles obtained, then taken a sentence that contains the word ambiguous and grouped into two files according to the category. From the file, the sentences that obtained from Wikipedia are then given a label to distinguish the meaning according to the group of ambiguous words. The flowchart of Wikipedia article selection can be seen in Figure 1.

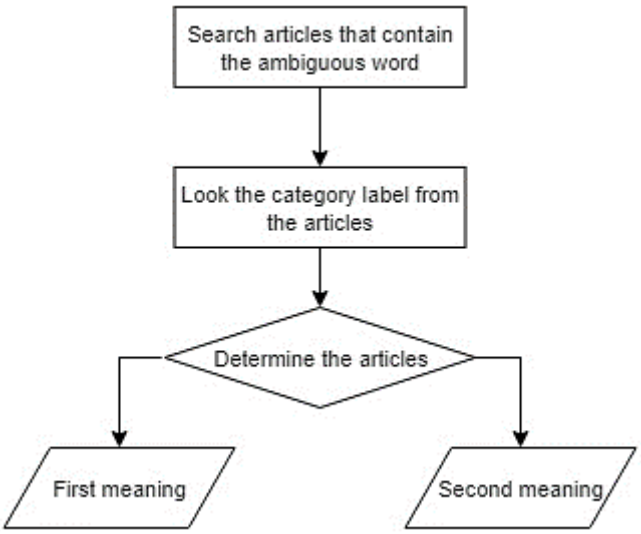


Figure 1. Wikipedia article selection

### *B. Training and Testing data*

From the search results on the Wikipedia site obtained a sentence that contains the ambiguous word and has been labeled. This label is used to distinguish the meaning of the ambiguous word. Having obtained a sentence that contains the word ambiguous the next list of sentences is used as data training that will be used in machine learning. The use of Wikipedia as training data because Wikipedia is known as a site that has high editorial quality [11] and has a category that can be used as a label.

While as testing data searched from the article some internet sites such as from news sites and special sites to discuss articles that contain the above ambiguous words. From the site of the site, as well as on the training, then sentence sentences containing the word ambiguous "motor" and "bulan". The sentences then labeled as in the training data. Labeling on the data testing is done manually. This label will not be processed in machine learning but will be used as a comparison of machine learning process prediction results.

### *C. Preprocessing*

The training data that obtained from Wikipedia is then preprocessed to eliminate the noise word of the sentence which has no significant meaning in the sentence [15]. The preprocessing stage is:

*1) Lowercase:* This process is done by changing all words into lowercase. This is so that each word is unique without any distinction between uppercase and lowercase.

*2) Remove punctuation*: Because punctuation is considered to have insignificant meaning in the sentence, so it needs to be eliminated.

*3) Stemming:* This process is done by changing the existing word into a basic word, especially for the verb.

*4) Remove stop words:* This process is done by eliminating words that have no significant meaning in sentences such as words that, and, others and the like.

Training data is done by preprocessing process with 4 stages above because the data is taken from good sources in language review.

For the testing data is still done additional stage preprocessing, because the search results of data from multiple sites allows there are data that have words that are not standard or contain abbreviations. Here's an additional process of preprocessing in data testing.

*5) Normalized slang word:* Slang word is a non-standard word used by people in unofficial communication. These words need to be matched with the standard word.

*6) Normalized abbreviation word*: Normalization of words containing abbreviations is only done on words that have meaning and influence in the sentence. While words that have no meaning and influence in the sentence will be deleted.

The example of preprocessing result can be seen in Table 1.

TABLE 1. PREPROCESSING RESULT

| Input | Output |
|---|---|
| saya pergi ke bulan | pergi bulan |
| saya menaiki sepeda motor | naik sepeda motor |

*D. Machine Learning using SVM*

Data that has been preprocessed then will be processed in machine learning with stages as below

*1) TF-IDF Feature Extraction:* Term Frequency-Inverse Document Frequency is a feature selection that will change the word in the document into a value (weight) in a statistic where the weight indicates how important the word is in a document [16]. The Stages performed in the weighting by using the TF-IDF are as follows:

*a) Tokenization:* Preprocessing training data is transformed into uni-gram and bi-gram. Data used as uni-gram is meant to know the weight of each word. While made bi-gram to know the weight when two words have a related meaning.

Examples of uni-gram and bi-gram tokenization:
Sentence: bertemu bulan januari
Uni-gram: {bertemu, bulan, januari}
Bi-gram: {bertemu bulan, bulan januari}
Merge: {bertemu, bulan, januari, bertemu bulan, bulan januari}

The flowchart of tokenization process shown in Figure 2.

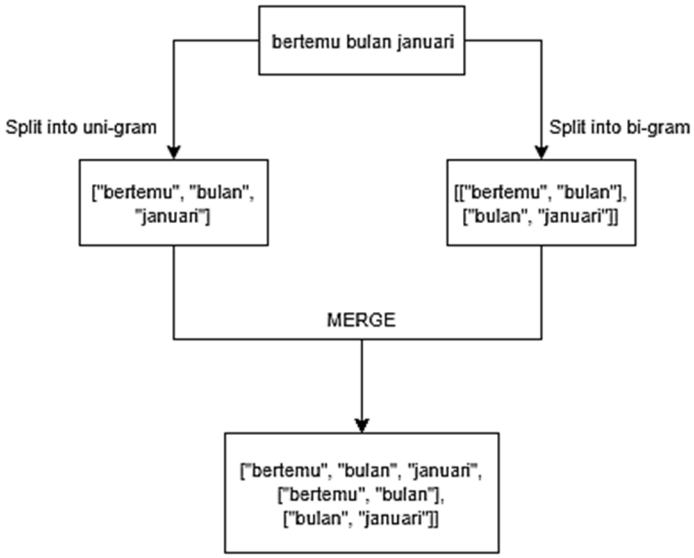


Figure 2. Tokenization

*b) Term Frequency:* Term frequency is done by counting the number of words that appear in a document [12]. The word that often appears indicates that the word is important. Term Frequency is calculated using the formula:

$$TF_{(t,d)} = \frac{n_{t,d}}{n_d} \qquad (1)$$

Formula (1), $TF_{(t,d)}$ is the number of terms t in the document $n_{t,d}$ is the number of occurrences of term t in document d. Whereas $n_d$ is the sum of the terms in the document d

*c) Inverse Document Frequency:* Inverse document frequency is a measure that determines how important a word is in all documents. It shows often or rarely a word appears in all documents. From this it is seen that preprocessing becomes important, as there are several words that very often appear but have no meaningful meaning like and, which, as well as, etc.

For the calculation of IDF in this paper will be used the formula:

$$IDF(t,d) = \log\left[\frac{(1+n)}{(1+df(t,d))}\right] + 1 \qquad 2)$$

From the above formula (2) it is known that *IDF (t,d)* is an inverse document frequency of term t in document d. *n* is the number of documents used, whereas $df(t,d)$ is the number of documents that contain term t. The addition of 1 of the formula (2) is intended for the sum to not be 0 to avoid the divisor 0.

While constant 1 is meant that the result of idf is at least 1.

After calculating the value of TF and IDF then calculated the value of TF-IDF with the following formula:

$$TF - IDF_{(t,d)} = TF_{(t,d)} . IDF(t,d) \qquad 3)$$

*d) Normalize TF-IDF:* Normalization is done using the L2 norm so that the weight of tf-idf for each term has a scale weight of 0-1, see the following formula

$$v_{norm} = \frac{v}{\|v\|^2} = \frac{v}{\sqrt{{v_1}^2 + {v_2}^2 + \cdots + {v_n}^2}} \qquad 4)$$

The example of TF-IDF weighting can be seen in Table 2.

TABLE 2. TF-IDF WEIGHTING

| No | Term | TF | | | | TF-IDF | |
|---|---|---|---|---|---|---|---|
| | | K1 | K2 | DF | IDF | K1 | K2 |
| 1. | temu | 1 | 0 | 1 | 1.17609 | 1.176090 | 0 |
| 2. | bulan | 1 | 1 | 2 | 1 | 1 | 1 |
| 3. | januari | 1 | 1 | 2 | 1 | 1 | 1 |
| 4. | temu bulan | 1 | 0 | 1 | 1.17609 | 1.176090 | 0 |
| 5. | bulan januari | 1 | 1 | 2 | 1 | 1 | 1 |
| 6. | berangkat | 0 | 1 | 1 | 1.17609 | 0 | 1.17609 |
| 7. | berangkat bulan | 0 | 1 | 1 | 1.17609 | 0 | 1.17609 |

*2) Suppor Vector Machine (SVM):* Support vector machine is a machine learning model used to predict classification analysis and regression analysis [17]. To

perform the classification, SVM will search for the best hyperplane function as a separator between two data classes.

In this research, we will use linear kernel function of SVM with the consideration that the data to be classified can be separated by a hyperplane.

### E. Performance Measure

The performance measure is determined by 4 (four) formulas: precision, recall, F1Score and accuracy.

Precision is a measure of the proximity of the correct ambiguous sentence prediction to the overall outcome of the correct ambiguous sentence. Precision can be written in the following mathematical sentences

$$Precision = \frac{Tx}{Tx + Fx} \quad (6)$$

Recall is the correct number of ambiguous sentence predictions versus the total number of ambiguous sentences in document x. Recall can be written in the following mathematical sentences

$$Recall = \frac{Tx}{Tx + Fy} \quad (7)$$

F1Score is a measure of test accuracy by considering the precision and recall results. F1Score can be written in the following mathematical sentence

$$F1\ score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (8)$$

The accuracy of the test is the ability to distinguish the correct ambiguous sentence predictions from the whole sentence tested. Accuracy can be written in the following formula:

$$Accuracy = \frac{Tx + Ty}{Tx + Fx + Ty + Fy} \quad (5)$$

where
$Tx$ = the amount of true prediction of the first sense
$Fx$ = the amount of false prediction of the first sense
$Ty$ = the amount of true prediction of the second sense
$Fy$ = the amount of false prediction of the second sense

For example, the word "bulan" has two senses, the first sense is "time" and the second sense are "earth satellite". We calculate for each sense. When the first sense is "time" then the second sense is "earth satellite" and vice versa.

## III. RESULT

In this paper, we use python as a programming language to implement the method we propose. For preprocessing Indonesian documents, we use Pysastrawi python library and for TF-IDF and SVM we use scikit-learn python library. Pysastrawi is a python library that allows us to reduce inflected words in Bahasa Indonesia to their base form, for example, the word "menahan" will be changed to "tahan". We use two ambiguous words as the testing data; first is "motor" with sense vehicle and propulsion (electric); second is "bulan" with sense time and earth satellite.

The confusion matrix result for "motor" can be seen in Table 3 and for "bulan" in Table 4. The performance measure of this experiment can be seen in Table 6.

TABLE 3. CONFUSION MATRIX OF "MOTOR"

| Prediction | Actual | |
|---|---|---|
| | Vehicle | Propulsion |
| Vehicle | 63 | 11 |
| Propulsion | 2 | 51 |
| Total Data | 65 | 62 |

TABLE 4. CONFUSION MATRIX OF "BULAN"

| Prediction | Actual | |
|---|---|---|
| | Time | Earth Satellite |
| Time | 46 | 7 |
| Earth Satellite | 8 | 43 |
| Total Data | 54 | 50 |

TABLE 5. EXPERIMENT RESULT

| Result | Bulan | | Motor | | Average |
|---|---|---|---|---|---|
| | Time | Satellite | Vehicle | Propulsion | |
| ***Precision*** | 0.852 | 0.860 | 0.969 | 0.823 | **0.876** |
| ***Recall*** | 0.868 | 0.843 | 0.851 | 0.962 | **0.881** |
| ***F1 score*** | 0.860 | 0.851 | 0.906 | 0.887 | **0.876** |
| **Accuracy** | 0.897 | 0.897 | 0.855 | 0.855 | **0.877** |

## IV. CONCLUSION

The methodology that we propose to solve word sense disambiguation in Bahasa Indonesia using Wikipedia and Support Vector Machine reach an accuracy rate of 0.877. This accuracy is good considering that there is no other method proposed for word sense disambiguation in Bahasa Indonesia. For further research, the senses of ambiguous words can be added so that not only have two senses.

## REFERENCES

[1] B. S. Rintyarna, R. Sarno, and C. Fatichah, "Enhancing the performance of sentiment analysis task on product reviews by handling both local and global context," *Int. J. Inf. Decis. Sci.*, vol. 11, 2018.

[2] E. Gabrilovich and S. Markovitch, *Journal of artificial intelligence research JAIR : an international electronic and print journal.*, vol. 34, no. 1. AI Access Found, 2009.

[3] A. Saif, M. J. Ab Aziz, and N. Omar, "Reducing explicit semantic representation vectors using Latent Dirichlet Allocation," *Knowledge-Based Syst.*, vol. 100, pp. 145–159, May 2016.

[4] A. Saif, N. Omar, U. Z. Zainodin, and M. J. Ab Aziz, "Building Sense Tagged Corpus Using Wikipedia for Supervised Word Sense Disambiguation," *Procedia Comput. Sci.*, vol. 123, pp. 403–412, Jan. 2018.

[5] O. Acikgoz, A. T. Gurkan, B. Ertopcu, O. Topsakal, B. Ozenc, A. B. Kanburoglu, I. Cam, B. Avar, G. Ercan, and O. T. Yildiz, "All-words word sense disambiguation for Turkish," in *2017 International Conference on Computer Science and Engineering (UBMK)*, 2017, pp. 490–495.

[6] S. Singh and T. J. Siddiqui, "Sense Annotated Hindi Corpus," in *2016 International Conference on Asian Language Processing (IALP)*, 2016, pp. 22–25.

[7] D. Wali and N. Modhe, "Word Sense Disambiguation Algorithms in Hindi," 2015.

[8] J. Sarmah and S. K. Sarma, "Word Sense Disambiguation for Assamese," in *2016 IEEE 6th International Conference on Advanced Computing (IACC)*, 2016, pp. 146–151.

[9] R. Wongso, F. A. Luwinda, B. C. Trisnajaya, O. Rusli, and Rudy, "News Article Text Classification in Indonesian Language," *Procedia Comput. Sci.*, vol. 116, no. C, pp. 137–143, 2017.

[10] R. Navigli and Roberto, "Word sense disambiguation," *ACM Comput. Surv.*, vol. 41, no. 2, pp. 1–69, Feb. 2009.

[11] S. Tonelli, C. Giuliano, and K. Tymoshenko, "Wikipedia-based WSD for multilingual frame annotation," *Artif. Intell.*, vol. 194, pp. 203–221, Jan. 2013.

[12] 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, Mar. 2017.

[13] B. S. Rintyarna and R. Sarno, "Adapted weighted graph for Word Sense Disambiguation," in *2016 4th International Conference on Information and Communication Technology (ICoICT)*, 2016, pp. 1–5.

[14] K. Mishina, S. Tsuchiya, and H. Watabe, "Word sense disambiguation of adjectives using dependency structure and degree of association between sentences," in *2017 International Conference on Asian Language Processing (IALP)*, 2017, pp. 342–345.

[15] S. Vijayarani, M. R. Janani, and A. Professor, "TEXT MINING: OPEN SOURCE TOKENIZATION TOOLS – AN ANALYSIS," *Adv. Comput. Intell. An Int. J.*, vol. 3, no. 1, pp. 37–47, 2016.

[16] H. C. Wu, R. W. P. Luk, K. F. Wong, and K. L. Kwok, "Interpreting TF-IDF term weights as making relevance decisions," *ACM Trans. Inf. Syst.*, vol. 26, no. 3, pp. 1–37, Jun. 2008.

[17] C.-W. Hsu, C.-C. Chang, and C.-J. Lin, "A Practical Guide to Support Vector Classification."