QUESTION CATEGORIZATION USING LEXICAL FEATURE IN OPINI.ID

Christian Eka Saputra[1], Derwin Suhartono[2], and Rini Wongso[3]

[1,2,3]Computer Science Department, School of Computer Science, Bina Nusantara University
Jln. K. H. Syahdan No. 9, Jakarta Barat 11480, Indonesia
[1]christian.saputra001@binus.ac.id; [2]dsuhartono@binus.edu; [3]rwongso@binus.edu

Received: 22nd September 2017/ **Revised:** 28th November 2017/ **Accepted:** 4th December 2017

Abstract **-** This research aimed to categorize questions posted in Opini.id. N-gram and Bag of Concept (BOC) were used as the lexical features. Those were combined with Naïve Bayes, Support Vector Machine (SVM), and J48 Tree as the classification method. The experiments were done by using data from online media portal to categorize questions posted by user. Based on the experiments, the best accuracy is 96,5%. It is obtained by using the combination of Bigram Trigram Keyword (BTK) features with J48 Tree as classifier. Meanwhile, the combination of Unigram Bigram (UB) and Unigram Bigram Keyword (UBK) with attribute selection in WEKA achieves the accuracy of 95,94% by using SVM as the classifier.

Keywords: text classification, Bag of Concept, Naïve Bayes, Support Vector Machine (SVM), J48 Tree

I. INTRODUCTION

In this modern era, the number of information has increased massively in the form of text or multimedia (sound, images, video, etc.). As the fundamental form of data, the researchers highlight text which has been used for many tasks such as question answering system (Jovita *et al*., 2015), argumentation classification (Desilia *et al*., 2017), and recommender system (Gunawan, Tania, & Suhartono, 2016).

The evolution of information has also led to information overload. Some of the information seems meaningless now, but they can be a useful thing in the future. One of the best solutions is to use the process of categorization or classification of information. In text categorization, feature extraction method and machine learning algorithm strongly affect categorization accuracy.

According to Ikonomakis, Kotsiantis, and Tampakas (2005), one of the solutions that can be offered to face problems of massive information is to make the process of automatic text classification. Automatic text classification is needed as the number and varieties of text or multimedia information has grown massively and often unstructured. Thus, it becomes less useful if it is not treated properly.

From the business standpoint, the information gathered can be a benchmark and a good guideline in determining a company's policies or changing business processes to answer public's needs. If the information can be grouped well, the decision making can create the best solution. For example, business leaders can get proper information regarding their needs if the news classification is defined properly.

In the process of automatic text classification or information, some classifiers that can be used are Naïve Bayes, Support Vector Machine (SVM), and Decision Tree. Many linguistic researchers implement these algorithms in their research as they perform well. Other than classifier, features are very important to describe the characteristics of information from various viewpoints. Structural features, lexical features, syntactic features, and contextual features are defined as group of features in specific task (Stab & Gurevych, 2014).

One specific feature which quite succeeds in describing the meaning of one sentence is lexical feature. The lexical feature is a representation of indicators that have been defined previously and associated with the word, lexeme, and vocabulary. The word is not tied to any other words. The example of the implementation of lexical features that are used is the N-gram (unigram, bigram, trigram), Bag of Word (BOW), and Bag of Concepts (BOC).

There are several researches that have been conducted and associated with automation process of text classification using lexical feature. The test accuracy is obtained by comparing the implementation of lexical feature. The combination of N-gram, BOW, and Bag of stemmed Word is also used (Rahmoun & Elberricihi, 2007). The process undertaken is to use the corpus of test data derived from two sources of data. Those are Reuters and Newsgroups. The results reveal that it is the best representation in determining the classification of a text derived from N-gram compared with other features such representations by BOW or Bag of stemmed Word.

Wei *et al.* (2008) used N-gram feature in Mandarin text. They also used a big corpus from TanCorp. It consisted of more than 14.000 texts and was divided into 12 classes. They mentioned the advantages of using N-gram were no word segmentation, and no special techniques and dictionary required for the implementation. The experiments concluded that bigram is the best feature for Mandarin. The experiments also implemented the combination of N-gram with 1-, 2-, 3-, and 4-gram that gave the best result, followed by 1-, 2-gram, 2-gram. The worst feature was by using only 1-gram. Mandarin mostly consisted of only 1 or 2 characters. Some of the Chinese scientifics' names consisted of more characters that made the combination of N-gram gave a good result in text classification process.

Sahlgren and Coster (2004) used a new approach to represent new feature in text categorization. They utilized BOC that combined some words with similar meaning. The result from BOC was compared to the result from BOW. It only calculated the frequency of occurrence of a feature derived from each word of documents.

The new approach by Sahlgren and Coster (2004), BOC or concepts based representation, is considered to be more efficient and fast. Additionally, it does not require additional external resources. Random indexing is also used in the implementation of BOC which aims to accelerate process of giving values for vector space model. It is due to expensive cost of BOC regarding computational cost. The experiment concluded that BOW (82,77%) gave good result for a small number of documents using Linear Kernel and TF-IDF rather than BOC (82,29%) with Polynomial Kernel and TF-IDF. However, in a large number of documents such as REUTERS-21578, BOC performed better (88,74%) compared to BOW (88,09%) (Sahlgren & Coster (2004).

Other researchers classify text on a biomedical literature. The classification process uses supervised learning method. The classifier processes the data with a few samples that have had previous categories. Both apply a data model formed into a set of other documents that serve as test data. The research compares the advantages and disadvantages of using a system of BOW and BOC in the process of transformation of a feature in the Vector Space Model. Researchers agree that the concept of BOW has high level of sparse data to produce the high dimensionality of the data. Therefore, the researchers use BOC in the transformation process feature. This concept explains about unit of meaning which means the unity of the various meanings (Garcia, Rodriguez, & Anido, 2015).

Moreover, the classification process of information is very useful to facilitate the formation of new meanings. It also can be the representations of data into useful information for the future.

According to Movementi (2015), one of a popular news portal in Indonesia, Opini.id is the combination of social media and news portal. Opini.id is a portal that facilitates Indonesians and the communities in giving opinions and sharing. Its mission is to facilitate and supports public opinions in developing Indonesia. In this portal, Indonesians are free to share their thoughts and opinions. Therefore, a lot of information can be obtained through this portal. However, many opinions are posted in Opini.id. Therefore, it is difficult for admins to manually categorize it to manage or analyze the data. This can lead to incorrect labeling.

Based on the problem, this research aims to categorize questions posted in Opini.id (Opini.id uses Indonesian language). The researchers will use lexical features of N-gram and BOC with Naïve Bayes, SVM, and Decision Tree (J48 Tree) as the classifier using Java WEKA API.

II. METHODS

In this research, the posted questions in Opini.id will be processed for classification. It will use the lexical features and three classifiers of Naïve Bayes, SVM, and Decision Tree (J48 Tree). There are two main phases in the automation of text classification. The phases are data preprocessing and data processing. The first step is data preprocessing. Data preprocessing is divided into three main processes. They are streaming data, stop word removal, and stemming. Figure 1 describes steps in data preprocessing.

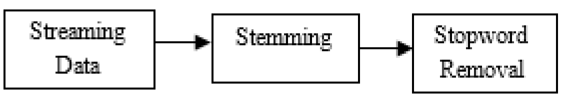

Figure 1 Data Preprocessing Steps

Streaming data is the initial process to obtain data so it can be used in the text classification process. The data source is derived from internal data of Opini.id. It has 12.700 rows in a single database and has been divided into ten main categories. The categories are business, technology, sports, health, travel, politics, celebrities, lifestyle, art, and education.

The second step of data preprocessing is stemming. It is to stem all data used as training data by using stemming algorithm proposed by Nazief and Adriani (1996). The algorithm works by using the rules of Indonesian morphological words. It removes the prefix and suffix of a word and uses the stem words database provided previously to decide the stemming process. This algorithm emphasizes the use of stem word database. Hence, the more complete list of words provided is, the higher the accuracy results will be.

The enhanced implementation of stemming process in this research is by doing stemmed word storage mechanism. The words that have passed through stemming is stored in a database. It aims to reduce the long computing process by checking and taking existing result from stemmed word database. The process has already been defined previously without the needs of repeating the stemming process from beginning. This enhancement provides an excellent time and efficiency in stemming process. Thus, it can be directed through the primary process, and the creation of training data and data model in text classification. In this research, stemmed words are stored in database as a reference for the next stemming process. Hence, base words will be taken from the database for the same words that have been previously stemmed.

The last step of data preprocessing is stopword removal. It will do word removal for certain word types such as conjunction word (“*dan*”, “*atau*”, and others) and interjection word (“*duh*”, “*wow*”, “*wah*”, and others). The stopwords list are taken from previously published research. The stopword removal is done to reduce noise in data to improve the computation. In this process, the statement like “*Ternyata memang Andi suka sepakbola sejak lama*” (literally means: evidently, Andi loves football since long time ago) will become “*Andi suka sepakbola*” (literally means: Andi loves football).

Stopword removal aims to improve the accuracy of automation classification process because the process of grouping into predetermined categories will be carried out. All words that have no connection or directly related to that category will be eliminated from the corpus of available data. It can reduce the sparse data on the implementation of word matrices in machine learning algorithm that is yet to be performed. The next step after data preprocessing is data processing. It is described in Figure 2.

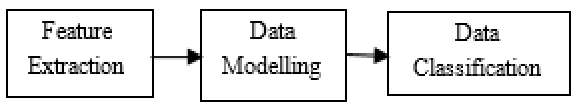

Figure 2 Data Processing Process

Feature extraction process is the process to get representation of data by extracting important characteristics from data. It needs some predefined categories of feature so that the extracted features can meet the actual needs. Lexical features used as features in this research are N-gram and BOC.

N-gram is a combination of words that can be obtained by stemming a longer string. The unique characteristic of an N-gram is that it is in a contiguous sequence of items of phonemes, syllables, letters, or words (Permadi, 2008). According to Hanafi, Whidiana, and Dayawati (2009), N-gram is a simple method for categorizing text or document with the superiority of not too sensitive with misspelling. However, by using N-gram, the feature matrices will become very huge.

N-gram implementation is not only in the form of character-based, but also word-based. N in N-gram indicates the size, unigram (N=1), bigram (N=2), and trigram (N=3). The example of implementation in N-gram can be seen in Table 1.

Table 1 N-Gram Implementation Example in Indonesian Language

	Word-Based	**Character-Based**
String	*"Andi suka bermain sepakbola di lapangan Senayan"* (literally means Andi loves to play football at Senayan stadium)	*"Pemerintah"* (literally means government)
Unigram	*Andi, suka, bermain, sepakbola, di, lapangan, senayan*	*p, e, m, e, r, i, n, t, a, h*
Bigram	*Andi suka, suka bermain, bermain ssepakbola, sepakbola di, di lapangan, lapangan senayan*	*pe, em, me, er, ri, in, ta, ah*
Trigram	*Andi suka bermain, suka bermain sepakbola, bermain sepakbola di, sepakbola di lapangan, di lapangan senayan*	*pem, eme, mer, eri, rin, int, nta, tah*

In this research, the construction of N-Gram is implemented by using internal data Opini.id. It is done by performing data retrieval from a database. Then, it implements a function to form N-gram as unigram, bigram, and trigram. Next, it is saved in a new data table.

Feature representation method with the concept of BOC is the new development from the previous transformation concept, BOW. This feature representation focuses on the meaning which contains a word. It is a combination of some words in a document which has the same meaning.

The concept of this feature transformation can be implemented by calculating the sum of all existing vector values. The values are based on the number of words in each document. The concept of BOC according to Täckström (2005), is proven to be implemented well in the information retrieval system, although this concept is quite simple. However, this concept has the disadvantage that is similar to BOW. The relevance or contextual relationships that exist in a word is not taken into account at all.

According to Sahlgren and Coster (2004), BOC can improve the performance of a classifier. In their research, the classifier tested was SVM. The experimental score increased from 88,74% to 88,99%. Despite the admittedly small differences, they insisted that it could not be negligible as it was consistent with the previous findings.

BOC in this research is implemented by collecting all internal data of Opini.id. Then, the data are put into Java Programming function that will create separation of some statement from those data into a few of words. It will be classified into several groups according to internal data of Opini.id categories.

The next step is data modeling. The purpose of this data model is to be the benchmark data for the new data that has not had a specific label. Then, the new data can be classified by specific category labels.

III. RESULTS AND DISCUSSIONS

The method proposed in the research is implemented using WEKA API .WEKA is a collection of machine learning algorithms for data mining tasks. As mentioned previously, this API is implemented in Java Spring framework.

The process of building data model starts with instances and attributes initialization. The term of instance is used as a condition of format or template in before the implementation of WEKA function. Instances are useful to accommodate training and testing data set. The attribute is used as a parameter feature in the categorization process. Figure 3 describes the process.

There are various types of data facilitated by the function and method of WEKA API. In this research, the numeric (representation regarding numbers) and class (representation in the form of selection category) are used. There is container initialization of dataset by using DenseInstance function. The example of a instance of dataset is 1, 2, 1, 3, 2, 5, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 4, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, Bisnis. Then, the data are inserted into instances in WEKA to create the data model. The data is built and ready for classification.

In this research, data classification is done using three classifiers of Naïve Bayes, Decision Tree (J48 Tree), and SVM with 10 categories of *"Bisnis", "Teknologi", "Olahraga", "Kesehatan", "Wisata", "Politik", "Selebritas", "Gaya Hidup", "Seni"*, and *"Edukasi"*. The Naïve Bayes classifier is based on the Bayes rule of conditional probability. It makes use of all the attributes contained in the data. It also analyzes them individually as if they are equally important and independent of each other (Wongso et al., 2017). Decision Tree is a predictive machine learning model. It decides the target value of a new sample based on various attribute values of the available data. The internal nodes of a Decision Tree denote the different attributes. The branches between the nodes tell the possible values that these attributes may have in the observed samples. Meanwhile, the terminal nodes describe the final value (the classification) of the dependent variable. As described by Ozer (2008), C4.5 algorithm is an algorithm that can be used to generate a Decision Tree developed by Ross Quinlan. Meanwhile, J48 Tree is an open source Java of C4.5 algorithm in the WEKA data mining tool. The algorithm uses a technique to induce Decision Tree for 20 classifications and uses reduced-error pruning (Ozer, 2008). The WEKA tool provides many options associated with tree pruning. In case of potential overfitting pruning, it can be used as a tool for précising. In other algorithms, the classification is performed recursively till every single leaf is pure, so the classification of the data should be as perfect as possible. This algorithm generates the rules which particular identity of that data is generated. The objective is to generalize Decision Tree progressively until it gains equilibrium of flexibility and accuracy (Kaur & Chhabra, 2014). According to Nugroho, Witarto, and Handoko (2003), SVM is a method used for pattern recognition process. SVM is a machine learning algorithm with structural risk minimization. It aims to search for a hyperplane by separating two classes of data in an input space. Hyperplane can be measured by a margin or distance. The nearest pattern to the borderline of a hyperplane is called as support vector.

The classification is carried out after the formation of the data model. It is derived from the data that have been trained. It begins with obtaining data on instances to do the categorization as described in Figure 4.

Then, categorization is performed based on the existing data model. The prediction of categorization can be obtained along with possible value acquisition for each data. The whole process is described in Figure 5.

The comparison process of data model testing uses the cross-validation with n = 10 with the classifier of Naïve Bayes, SVM, and J48 Tree are described in Table 2 and Table 3. The feature combinations use the cross-validation folds = 10. Based on that, the highest accuracy of 96,5% is obtained by using features of Bigram Trigram Keyword (BTK). It uses J48 Tree classifier which outperforms the other classifiers and the other feature combinations. The highest accuracy after BTK and J48 Tree is obtained by using Bigram Keyword (BK) and J48 Tree. It is with the accuracy of 96,41%. It is slightly lower than the previous one.

Meanwhile, the worst result is obtained by using feature of Unigram Keyword (UK) with Naïve Bayes as the classifier. It only obtains 41,79%. It is far below the average. The results by using feature of UK consistently give the worst result among other features. It is with 51,92% of accuracy using J48 Tree, and 62,72% of accuracy using SVM.

```
ArrayList<Attribute> attribute arff = new ArrayList<Attribute>();
ArrayList<String> nama_kategori = new ArrayList<String>();

nama_kategori.add("Bisnis");
nama_kategori.add("Teknologi");
nama_kategori.add("Olahraga");
nama_kategori.add("Kesehatan");
nama_kategori.add("Wisata");
nama_kategori.add("Politik");
nama_kategori.add("Selebritas");
nama_kategori.add("Gaya Hidup");
nama_kategori.add("Seni");
nama_kategori.add("Edukasi");

attribute_arff.add(new Attribute("feature_keyword_category1));
attribute_arff.add(new Attribute("feature_keyword_category2));
attribute_arff.add(new Attribute("feature_keyword_category3));
attribute_arff.add(new Attribute("feature_keyword_category4));
attribute_arff.add(new Attribute("feature_keyword_category5));
attribute_arff.add(new Attribute("feature_keyword_category6));
attribute_arff.add(new Attribute("feature_keyword_category7));
attribute_arff.add(new Attribute("feature_keyword_category8));
attribute_arff.add(new Attribute("feature_keyword_category9));
attribute_arff.add(new Attribute("feature_keyword_category10));

Instances trainset_arff = new Instances("Classification", attribute_arrf,
result.size());
Instances trainset_container = new Instances("Classification", attribute_arff, 0);
```

Figure 3 Build Data Model Process

```
Instance newInst =
testing_demo_container_instance(0);
```

Figure 4 Data Modelling Process

```
Naïve Bayes nb = new NaiveBayes();
nb.buildClassifier(trainset_container);
Double predNB = nb.classifyInstance(newInst);
String predString =
testing_demo_container.classAttribute().value(predNB.intValue());
double[] probabilities = nb.distributionForInstance(newInst);
```

Figure 5 Data Classification Process

Next, the other experiment is done by using the feature combinations with attribute selection and the classifier of Naïve Bayes, SVM, J48 Tree. The results are shown in Table 3.

Table 2 Experiment Using N-gram Features

Feature Combination	Classifier		
	Naïve Bayes	**J48 Tree**	**SVM**
UB	80,39	96,04	93,37
UT	80,81	91,3	88,89
UK	41,79	51,92	62,72
UBT	89,23	96,21	94,84
UBTK	79,79	91,02	86,18
UBK	89,48	96,04	92,7
BT	90,12	96,21	95,83
BK	90,43	96,41	95
BTK	90,26	96,5	95,07
TK	82,36	91,86	90,71
UTK	81,32	91,04	88,6

Description:
UB = Unigram Bigram
UT = Unigram Trigram
UK = Unigram Keyword
UBT = Unigram Bigram Trigram
UBTK = Unigram Bigram Trigram Keyword
UBK = Unigram Bigram Keyword
BT = Bigram Trigram
BK = Bigram Keyword
BTK = Bigram Trigram Keyword
TK = Trigram Keyword
UTK = Unigram Trigram Keyword

Table 3 Experiment with N-gram and Attribute Selection

Feature Combination	Classifier		
	Naïve Bayes	**J48 Tree**	**SVM**
UB	89,56	95,78	95,94
UT	82,09	91,09	91,28
UK	44,35	44,48	53,66
UBT	88,8	95,92	95,88
UBTK	82,1	91,1	91,3
UBK	89,56	95,78	95,94
BT	88,8	95,92	95,88
BK	88,28	94,44	94,76
BTK	88,8	95,91	95,88
TK	82,09	91,09	91,28
UTK	82,1	91,1	91,28

According to the results shown in Table 3, the best result is obtained by using feature of Unigram Bigram (UB) and Unigram Bigram Keyword (UBK) with classifier of SVM. The accuracy is 95,94% and is followed by UBT-J48 Tree with 95,92% of accuracy. Meanwhile, BTK-J48 Tree is with 95,91% of accuracy. The worst result is still achieved by using feature of UK and classifier of Naïve Bayes with only 44,35% of accuracy. It is followed by using J48 Tree (44,48%) and SVM (53,66%).

IV. CONCLUSIONS

Based on experiment comparison of automation classification process by using lexical feature implementation, it can be concluded. The researchers conclude that the experiments are carried out with a combination of lexical features between unigram, bigram, trigram, and keywords of each category in the implementation of data modeling. It uses cross-validation by the number of fold about 10. It shows that the combination of bigram, trigram, and keyword gives the highest accuracy of 96,5% with J48 Tree.

Moreover, the experiment with combination of lexical feature by using the attribute selection feature is done. It is to find out what are the most significant features that affect the process of automation category of questions. Based on the experiments, it can be seen that the combination of lexical feature UB and UBK by using SVM Classifier provides a high percentage compared to others. It is with accuracy of 95,94%.

For further utilization of this finding, the other features such as structural and contextual are suggested to be attached to the current features. This good result of using lexical features depicts that any classification or categorization problems should not ignore the importance of lexical knowledge from the texts. If there are bigger data for this task, deep learning is interesting to experiment as well. Based on the data characteristics, text and Long-Short Term Memory (LSTM) will be the best fit for this question categorization task.

REFERENCES

Desilia, Y., Utami, V. T., Arta, C., & Suhartono, D. (2017). An attempt to combine features in classifying argument components in persuasive essays. In 17th *Workshop on Computational Models of Natural Argument (CMNA).* London, United Kingdom.

Garcia, M. M., Rodriguez, R. P., & Anido, L. (2015). Bag of concepts document representation for textual news classification. *International Journal of Computational Linguistics and Applications, 6*(1), 173-188.

Gunawan, A. A. S., Tania, & Suhartono, D. (2016). Recommender system for product offering by personalized email. In *1st International Workshop on Big Data and Information Security (IWBIS)*. Jakarta, Indonesia.

Hanafi, A., Whidiana, R., & Dayawati, R. N. (2009). *Pengenalan bahasa suku bangsa Indonesia berbasis teks menggunakan metode N-Gram* (Skripsi). Bandung: Telkom University.

Ikonomakis, M., Kotsiantis, S., & Tampakas, V. (2005). Text classification using machine learning techniques. *WSEAS Transactions on Computers, 4*(8), 966-974.

Jovita, Linda, Hartawan, A. & Suhartono, D. (2015). Using vector space model in question answering system. *Procedia Computer Science, 59*, 305-311.

Kaur, G., & Chhabra, A. (2014). Improved J48 classification algorithm for the prediction of diabetes. *International Journal of Computer Applications, 98*(22), 13-17.

Movementi, S. (2015). *Opini.id unggulkan fitur polling*. Retrieved from https://tekno.tempo.co/read/

news/2015/02/26/072645583/opini-id-unggulkan-fitur-polling

Nazief, B., & Adriani, M. (1996). *Confixstripping: Approach to stemming algorithm for Bahasa Indonesia.* Jakarta: Faculty of Computer Science, University of Indonesia.

Nugroho, A. S., Witarto, A. B., & Handoko, D. (2003). Application of support vector machine in Bioinformatics. In *Indonesian Scientific Meeting in Gifu, Central Japan.*

Ozer, P. (2008). *Data mining algorithm for classification* (Bachelor Thesis). Redbound University Nijimegan

Permadi, Y. (2008). *Kategorisasi teks menggunakan N-Gram untuk dokumen berbahasa Indonesia* (Skripsi). Bogor: Institut Pertanian Bogor.

Rahmoun, A., & Elberricihi, Z. (2007). Experimenting N-Grams in text categorization. *The International Arab Journal of Information Technology, 4*(4), 377-385.

Sahlgren, M., & Coster, R. (2004). Using bag of concepts to improve the performance of support vector machines in text categorization. In *Proceedings of the 20th International Conference on Computational Linguistics Article No 487*.

Stab, C. & Gurevych, I. (2014). Identifying argumentative discourse structures in persuasive essays. In *Conference on Empirical Methods on Natural Language Processing (EMNLP).*

Täckström, O. (2005). *An evaluation of bag-of-concepts representations in automatic text classification* (Master Thesis). Swedia: Royal Institute of Technology Sweden.

Wei, Z., Miao, D., Chauchat, J. H., & Zhong, C. (2008). Feature selection on Chinese text classification using character N-grams. In *International Conference on Rough Sets and Knowledge Technology* (pp. 500-507). Springer.

Wongso, R., Luwinda, F., Trisnajaya, B., Rusli, O., & Rudy. (2017). News article text classification in Indonesian language. In *The 2nd International Conference on Computer Science and Computational Intelligence (ICCSCI 2017)* (pp. 137-143). Elsevier.