J Inf Process Syst, Vol.20, No.1, pp.1~12, February 2024
https://doi.org/10.3745/JIPS.02.0209

ISSN 1976-913X (Print)
ISSN 2092-805X (Electronic)

JOURNAL OF INFORMATION PROCESSING SYSTEMS JIPS

Semantic Feature Analysis for Multi-Label Text Classification on Topics of the Al-Quran Verses

Gugun Mediamer and Adiwijaya*

Abstract

Nowadays, Islamic content is widely used in research, including Hadith and the Al-Quran. Both are mostly used in the field of natural language processing, especially in text classification research. One of the difficulties in learning the Al-Quran is ambiguity, while the Al-Quran is used as the main source of Islamic law and the life guidance of a Muslim in the world. This research was proposed to relieve people in learning the Al-Quran. We proposed a word embedding feature-based on Tensor Space Model as feature extraction, which is used to reduce the ambiguity. Based on the experiment results and the analysis, we prove that the proposed method yields the best performance with the Hamming loss 0.10317.

Keywords

Text Classification, Tensor Space Model, The Al-Quran Verses, Word Embedding

1. Introduction

The rapid growth of technology in the digital world is producing enormous amounts of data and causing unstructured text data [1]. Text classification is one of the fields for processing unstructured text data into informative. Many researchers have conducted research related to text classification. For instance, the studies conducted by Pane et al. [2] and Kim et al. [3]. Every data used in text classification requires different treatments to obtain the best performance. Nowadays, the contents of Islam are widely used in text classification research, including Hadith and the Al-Quran. Hadith and the Al-Quran are used as the sources of law for Muslims in the world. Both are also used as guidelines for daily life by Muslim [4]. Previous studies related to text classification on Hadith have been done by Mediamer et al. [5] and Bakar and Al Faraby [6]. Both discussed the multi-label classification of Bukhari Hadith into three classes, namely prohibition, advice, and information. Mediamer et al. [5] prove that rule-based feature extraction and support vector machine (SVM) as a classifier could improve the accuracy of the classification system. While Bakar and Al Faraby [6] used information gain as a feature selection in the classification process with back propagation as a classifier, and the system was able to reduce running time and even increase the accuracy.

The Al-Quran is the main source of law for Muslim [4]. The Al-Quran consists of 6,236 verses which are grouped into 114 surah [7]. The Al-Quran readers are divided into two, which are Muslim and non-

Manuscript received July 5, 2021; first revision August 18, 2022; accepted September 29, 2022.
* **Corresponding Author:** Adiwijaya (adiwijaya@telkomuniversity.ac.id)
School of Computing, Telkom University, Bandung, West Java, Indonesia (adiwijaya@telkomuniversity.ac.id, mediamergugun@student.telkomuniversity.ac.id)

Muslim. Muslim use the Al-Quran as the guide and way of life. Based on the population report that has been conducted by Pew Research Center Religion & Public Life in 2015, the number of Muslim populations predicted to be 1.9 billion in 2020 [8]. It means, the Muslim population is 24.9% of the total population in the world. However, not every Muslims, or readers in general, can understand the Al-Quran easily, since in each verse of the Al-Quran might contain more than one topic. Therefore, the readers found limitation to determine the topics of the Al-Quran verses. On the other hand, non-Muslim also read the Al-Quran to expand their insight about the main source of law in Islam. Therefore, this research is proposed to help both Muslim and non-Muslim readers, especially for Muslim, to learn the Al-Quran easily in the digital form. This could be done by implementing a word embedding feature.

Verses of the Al-Quran can be classified into multi-label topics. The topics that contained in the Al-Quran can be grouped into 15 topics, such as Pillars of Islam, Faith, the Al-Quran, Science and its Branches, Working, Call to God, Jihad, Human and Social Relations, Morals, Regulations Related to Property, Legal Matters, State and Society, Agriculture and Trade, History and Stories, and Religions [2]. Therefore, the system of text classification to be built needs a process which can classify each verse of the Al-Quran into multi-label topics precisely.

Pane et al. [2] has conducted research related to the multi-label classification of verses of the Al-Quran. The research used bag-of-words as feature extraction and multinomial Naive Bayes as a classifier. However, bag-of-words does not focus on the semantics relationship between each term. Other than that, Izzaty et al. [9] and Ulumudin et al. [10], both have also conducted study related to multi-label classification on the Al-Quran verses. Izzaty et al. [9] proposed a tree augmented Naïve Bayes as a classifier method, while Ulumudin et al. [10] proposed k-nearest neighbor (KNN) with term frequency-inverse document frequency (TF-IDF) weighted. On the other hand, Huda et al. [11] proposed the neural network approach as a classifier with Adam optimizer in handling multi-label data on the Al-Quran verses. Meanwhile, Nurfikri [12] has done with the study to compare the performance between neural network approach and SVM on multi-label classification of the Al-Quran verses.

Previous research related to the text classification has also been done by Kim et al. [3]. The study classified news in English translations. One of the features used in the study is semantic concept features. The feature weight is obtained using the Lesk algorithm, which focuses on calculating the overlap between the definitions of the words [13]. This study describes a document into 2nd order-tensor (matrix) that is called semantic features. So as the semantic features are described as independent space in 3rd order-tensor of the dataset. The 3rd order-tensor is also called multilinear algebra, which is a generalization of the concept of vectors in a linear algebra field. The vector structure can be called as 1st order-tensor, the matrix structure as a 2nd order-tensor, and the cube-like as 3rd order-tensor, and so on [14].

This study proposed a semantic feature using a word embedding system. The aim of using a word embedding system is to classify multi-label based on topics of the Al-Quran verses in English translation and to produce a better performance of classification. The rest of this paper discusses the previous research related to multi-label classification, semantics features, and the classifier. The system design is explained in Section III. Then, the result and discussion are explained in Section IV, and finally, the conclusion of this study is presented in Section V.

2. Related Work

Several research related to the text classification in English translation has been done by many

researchers, because there are many open access data and easy to obtain. The verses of the Al-Quran are originally in Arabic. However, currently the verses are now also available in English translation. Some studies using the dataset of English translation have been performed [2,10-12]. In these data there are a lot of noise that can affect the performance of classification results. Therefore, it requires several stages in pre-processing to minimize the amount of data (the noise) and ready to be used in classification process [15].

Generally, dataset is represented by document-by-term matrix. Bag-of-words are the basic features commonly used in this representation. The weakness of this approach is the loss of the meaning of the word, because it only focuses on the frequency in which the terms appear in the document [3]. Kim et al. [3] proposed a 3rd order-tensor representation, which also called a tensor space model. Each document in the dataset is represented by the term-by-concept matrix, where in general the document is represented by the terms vector. Thus, the dataset will be represented as a document-by-term-by-concept tensor [3]. Several studies have used tensor space models [3,14,16-18]. All these studies show that the tensor space model can produce better classification than the baseline representation, because of tensor space model can used semantic features by representing matrix for each of documents.

In several previous studies, Lesk algorithm was used to build semantic features with concept based, including those conducted by Kim and his colleagues [3,14,16]. Basically, Lesk algorithm calculates overlapping on the definition of words contained by the dictionary [14]. Therefore, Lesk algorithm is very dependent on the definition of words available in the dictionary to calculate the overlapping. Dictionary is not always able to provide all the words, because the words will always increase, and it is not possible for the dictionary to be updated continuously. Furthermore, not all languages have a complete dictionary like English with its WordNet lexical dictionary.

The word embedding system has been conducted in several studies to handle semantic features. The representation using this approach has been shown to yield good classification results [5,19-21]. The studies used vector representation as the feature weighting method. Unlike another study [22], word embedding system was actually used for the feature selection process by clustering features based on similarities. Furthermore, the most important thing in the classification text is the classifier. The classifier is used to determine the class for each document. There are several classifier methods commonly used in text classification problems. For instance, Naïve Bayes, SVM, KNN, and neural network. Nevertheless, Kim et al. [3] proposed the classifier, which is semantic Naive Bayes classification, the classifier is a development of conventional Naive Bayes in order to be used on data with tensor space model [3].

3. Research Method

Fig. 1 explains how the system in this study was built. The Al-Quran verses in English translation taken from the study by Pane et al. [2]. For each verse of the Al-Quran has been given as many as 15 classes in accordance with the topics. There are several steps in the system to complete the task. First, the data processed by preprocessing step, this step has five sub-processes such as case folding, remove punctuation, tokenization, stopword removal, and lemmatization, to produce the clean data and the terms of document. Subsequently, the data split into two parts, such as train data and test data. Meanwhile, other process collects the Wikipedia corpus in accordance with the terms of document. After collecting, the Wikipedia corpus processed by cleaning process. So as the clean corpus used in extracting semantic

concept features. Training data processed to extract the semantic concept features by Lesk and word embedding. The next step is classification process, which is used to classify the testing data using trained data. Finally, the classification result processed by the evaluation process to calculate the accuracy of the classification. The following is a detailed explanation of the system being built.

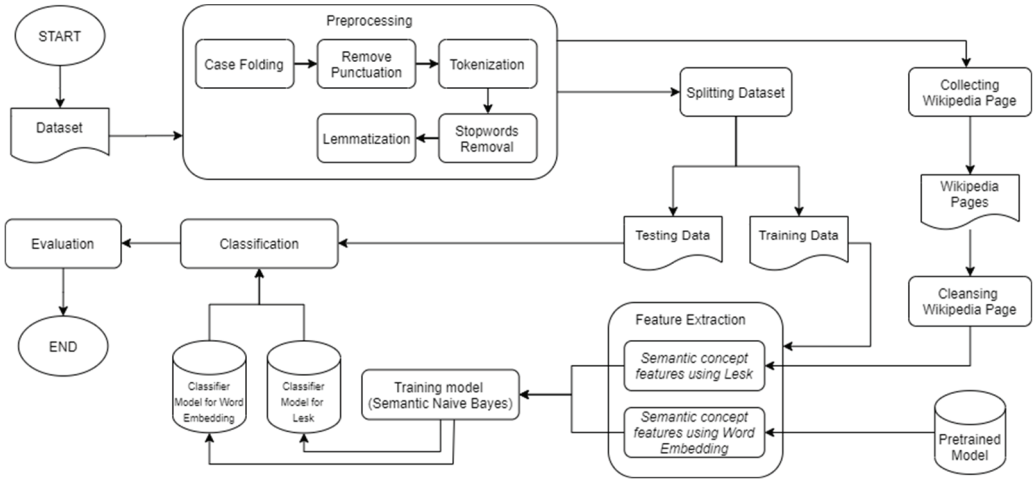


Fig. 1. Design of system.

3.1 Dataset

The dataset used in this study is the verses of the Al-Quran in English translation. The Al-Quran consists of 6,236 verses and each verse contains a different topic. The dataset was obtained from the Al-Quran publisher namely Syaamil Quran [23]. The study used 15 topics as a multi-label class for each verse. A list of topics is defined in the previous research that has been done by Pane et al. [2] as mentioned earlier.

There are three sample data in Table 1. The first verse in the Al-Quran is indicated by No. 1, in the 16 columns afterward it can be seen that the verse contains the topic "Pillar of Islam" marked with the value "1" in the column that corresponds to the order of the topic numbers. Then in the second verse, this verse does not contain all topics, therefore column "16" is worth "1" and the previous 15 columns are worth "0." If a verse contains more than one topic, then columns 1 through 15 can be valued at "1" simultaneously. The handling of the third paragraph and so on will always be the same as the first and second verse.

Table 1. Example of data

No.	Verse	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	In the name of Allah, the Beneficent, the Merciful.	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	All praise is due to Allah, the Lord of the Worlds.	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
3	The Beneficent, the Merciful.	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

3.2 Collecting Wikipedia Pages

This study used external corpus which is the Wikipedia pages to get the semantic concept. The Wikipedia pages are selected in accordance with the terms in the dataset, and these are processed in the feature extraction phase. As an example, one of the terms contain in a dataset is "Religion," so the system takes Wikipedia pages related to it as shown in Fig. 2.

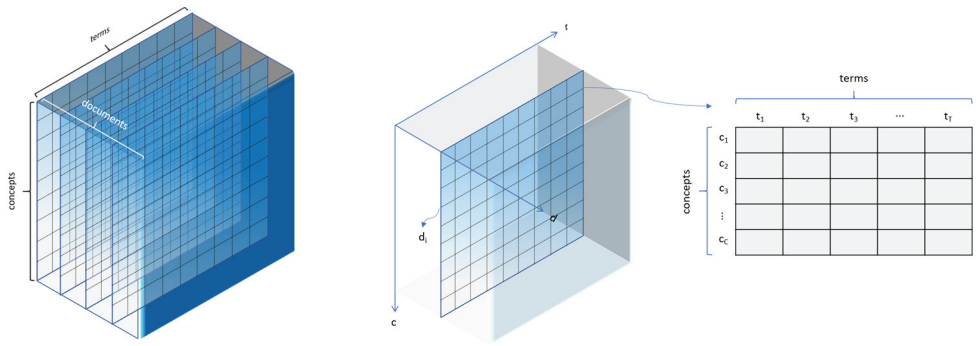
The system retrieves Wikipedia pages depending on the availability of corpus in the Wikipedia database. If the page with certain terms is not available in the Wikipedia corpus, then the term will be removed from the list terms. These terms are deleted because if they continue to be used, they will not be able to provide the semantic concept value, which is needed by the system. In addition, after the system has collected all required Wikipedia pages, the next step is the process of cleaning up Wikipedia pages. The Wikipedia page is cleaned using the preprocessing steps such as removing white space, newline, symbols, and non-alphabetic characters.

Fig. 2. Example of Wikipedia pages (https://en.wikipedia.org/wiki/Religion).

3.3 Feature Extraction

This study used two types of main features. The first is Lesk, which is used as the baseline method. The second is feature extraction with Word Embedding, which is the proposed approach in this study and will be compared to Lesk. The Lesk algorithm was explained clearly in previous researches [3,14]. In addition, we also conducted some experiments regarding these two types of features.

At the feature extraction stage, the system described document by 2nd order-tensor (matrix), mean the dataset described by 3rd order-tensor as has been done by research [3,24]. Fig. 3(a) illustrates the term-documents-concepts as data representation, which represent 3rd order-tensor. This study used a term-by-concept to describe a document. The terms produced by the dataset; terms mean the collection of unique words that occur in the dataset. While the concept is using an external Wikipedia corpus in accordance with the unique words. The representation for each document was illustrated in Fig. 3(b). The document is represented by a term-by-concept matrix. Concept means a collection of Wikipedia pages that match the terms. The size of the term-by-concept matrix is the total number of terms, times the total number of Wikipedia pages.



(a) (b)

Fig. 3. (a) The 3rd order-tensor and (b) as document representation as a term-by-concept matrix.

This study proposed several types of features on two main feature models. We build features using Lesk and Word Embedding in few methods. The first is feature of Lesk. This feature model is used as a baseline model for the classification process. Features are represented by a term-by-concept matrix. Fig. 4(a) describes the representation of Lesk features used in the classification process. In addition, Lesk has an important parameter called concept windows. This parameter is used when building a feature model. We used concept windows as many as 2 and 5 to produce two different types of Lesk features.

Furthermore, the next feature is Word Embedding system. This feature is used as a proposed model feature. In this research, we take how Lesk representing the data. Term in the document is represented using word embedding. The word embedding model is obtained from the pretrained model using Fasttext with a vector length of 300. Fig. 4(b) is feature representation of word embedding for a document.

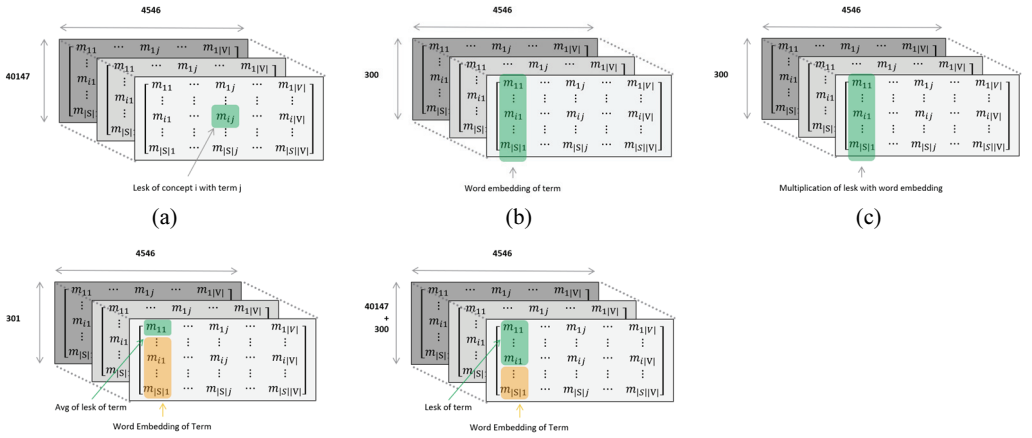


Fig. 4. Features extraction types: (a) Lesk, (b) word embedding, (c) Lesk multiply by word embedding, (d) average of Lesk concatenate with word embedding, and (e) Lesk concatenate with word embedding.

The rest of three features are the combination of Lesk and Word Embedding. The first is feature of Lesk Multiply by Word Embedding. The system multiplies each term-by-concept value for the Lesk feature with vector of word embedding of term. Then the average of vector is calculated based on the

terms. Therefore, the vector length for each term becomes 300. Fig. 4(c) is the matrix representation for a document. The next feature is feature of Average of Lesk Concatenate with Word Embedding. In this feature, system calculates the average for each Lesk of terms, then we combine it with the terms vector of word embedding. The matrix representation for each data is shown in Fig 4d. The last is feature of Lesk Concatenate with Word Embedding. The system combined the term-by-concept with the vector terms. The matrix representation for each data is depicted in Fig. 4(e). The disadvantage of this feature is the matrix dimension, which is the largest than other feature types.

3.4 Classification Method

This research used the semantic Naïve Bayes approach that proposed in the study by Kim et al. [3]. The advantage of this method is that the meaning for each term will be obtained with good semantic value by utilizing external corpus of Wikipedia pages [3]. The number of Wikipedia pages will increase according to the development of data used. So, it is possible that a term has no exact concrete meaning. Therefore, we need a smoothing method to deal with this. This study used the Laplace smoothing method as applied in [3].

$$\begin{aligned}\Phi_{\theta_{SNB}}(d) &= argmax_{c\in C}\,\Pr(c)\,.\Pr(d|c)\\ &= argmax_{c\in C}\,\Pr(c)\,.\prod_{i=1}^{|V|}\prod_{j=1}^{|S|}\Pr(m_{ij}|c)\\ &= argmax_{c\in C}\left\{log\Pr(c)+\sum_{i=1}^{|V|}\sum_{j=1}^{|S|}log\Pr(m_{ij}|c)\right\}.\end{aligned} \tag{1}$$

Eq. (1) [3] is semantic Naïve Bayes classification function for text documents, where $\Pr(c)$ is a class prior probability document from the corpus of documents with class c and $\Pr(d|c)$ is the probability of a document d in class c. In addition, the document d is classified as class $\Phi_{\theta_{SNB}}$ with the highest posterior probability value.

In Naïve Bayes approach, document d is represented by bag-of-words $(t_1, t_2, \dots, t_{|T|})$. Whereas in the semantic Naïve Bayes approach, document d is represented by the term-by-concept matrix as illustrated in Fig. 3. Prior probability value $\Pr(c)$ is the number of occurrences of document in the training set of documents with the class $c \in C$. In the semantic Naïve Bayes, $\Pr(d|c)$ is the probability of each cell $\Pr(m_{ij}|c)$ in the matrix term-by-concept illustrated in Eq. (2), where $|V|$ is the number of terms and $|S|$ is the number of concepts.

$$\begin{bmatrix} m_{11} & \cdots & m_{1j} & \cdots & m_{1|S|} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ m_{i1} & \cdots & m_{ij} & \cdots & m_{i|S|} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ m_{|v|1} & \cdots & m_{|v|j} & \cdots & m_{|V||S|} \end{bmatrix}, \tag{2}$$

$$\hat{p}r(m_{ij}|c) = \frac{w^*(t_{i,}\, s_{j,}\, c)}{\Sigma_{t\in V}\,\Sigma_{s\in S}\, w^*(t,s,c)}. \tag{3}$$

Eq. (3) is used to get the probability for each cell in the matrix, where $w^*(t_{i,}\, s_{j,}\, c)$ denotes the weighted

value for the term's i^{th} and the concept j^{th} in the class $c \in C$. The detail calculation of weighted value denoted by Eq. (4) [3], where $|c|$ is the number of documents with the class $c \in C$.

$$w^*(t_i, s_j, c) = \frac{\Sigma_{d \in c}\, w(t_i, s_j, c)}{|c|}. \quad (4)$$

The number of Wikipedia pages will increase according to the development of data used. So, it is possible that a term has no exact concrete meaning. Therefore, we need a smoothing method to deal with that problem, this study used the Laplace smoothing method as in [3]. Eq. (5) is used to estimate the cell probability for each class by adding the Laplace smoothing,

$$\hat{p}r(m_{ij}|c) = \frac{w^*(t_i, s_j, c) + 1}{\Sigma_{t \in V}\, \Sigma_{s \in S}\, w^*(t, s, c) + |V|\,|S|}. \quad (5)$$

4. Experimental Results and Analysis

In this study, we used Hamming loss to calculate the classification error of labels in the prediction results. For example, a data that should be predicted as labels 1 and 3, where the system predicts the data into labels 1 and 2. Furthermore, the Hamming loss is close to zero indicating that the system is almost perfect, and the system has the best performance if the result $HammingLoss(C, D) = 0$.

The first testing process is carried out to measure system performance when using Lesk as a feature model. The test results are numbers 1 and 2 as shown in Table 2. We built a feature model with different concept window parameter, such as the concept window as many as 2 and 5. Based on Table 2, a system with a Lesk feature model produced poor performances among other features. This is proven by producing 2 largest Hamming loss compared to the other 7.

Table 2. Summary of experimental results

No.	Model of feature	Concept window	Length of term vector	Hamming loss
1	Lesk	2	40,147	0.48803
2	Lesk	5	40,147	0.48803
3	Word embedding	-	300	0.10847
4	Lesk multiply by word embedding	2	300	0.10847
5	Lesk multiply by word embedding	5	300	0.10847
6	Average of Lesk concatenate with word embedding	2	301	0.10567
7	Average of Lesk concatenate with word embedding	5	301	0.10317
8	Lesk concatenate with word embedding	2	40,447	0.47055
9	Lesk concatenate with word embedding	5	40,447	1.45177

Furthermore, there is a difference in the Hamming loss that generated by changing the value of concept window. The models built with a length of concept window 5 denotes better results compared with length of concept window 2. This proved that the greater of concept windows, the better result of Lesk method to capture the concept features.

For the next feature, testing is carried out to measure the system performance when using word embedding as a feature. This model is used to compare the Lesk feature model with the word embedding. We use pretrained fasttext model with a vector length of 300. Therefore, we only use one model for this feature type. Based on the Table 2, the test result is shown in number 3. The Hamming loss shows a significantly difference compared to the Lesk method. This happens cause of word embedding can provide better performance than Lesk for a multi-label dataset of the Al-Quran verses in English translation. In addition, word embedding produces better vector representation of terms, because the training process is carried out with very large data.

Basically, the word embedding model has a drawback which is out of vocabulary, especially for data of the Al-Quran verses. Because of the Al-Quran is the holy book for Muslim, and the content in it discusses specific things about Islam. Therefore, there is a big possibility of out-of-vocabulary (OOV). However, in the fasttext method, this problem can be handled by sub word of the model. The model can produce vectors for OOV, even though the vector result is less precise. That is better than giving a null value for the system to process.

The third testing is carried out to measure the system performance when using the Lesk multiply by-word embedding feature. The aims are to prove the effect of word embedding on Lesk features. Based on Table 2, Hamming loss does not show performance improvement from the previous feature. In addition, the value of concept windows also has no effect on system performance. This happens because the multiplication process causes the shape of the feature to be like the word embedding feature model. However, the Hamming loss is same as the word embedding feature model, but it is better than the stand-alone Lesk feature.

Another type of feature is the average of Lesk Concatenate with Word Embedding. This experiment is conducted to prove the effect of the average of Lesk feature concatenates with word embedding. Table 2 with numbers 6 and 7 shows interesting results. The Hamming loss shows smallest values compared to the other since the average feature of Lesk indicates the unique value of the vector terms. In this feature, the larger concept window yields a better result, because of getting a better representative meaning of terms.

The final testing process is carried out using the Lesk concatenate with word embedding model. The aim is to analyze the effect of word embedding vector concatenate with vector Lesk directly. Combined directly means to combine a vector of term for length of 40,147 Lesk features with a vector term word embedding with a length of 300. So that the length of the vector term becomes 40,447, and it is the longest vector of the term compared to other types of features. It is also can improve computing time for systems in the learning process.

In addition, in Table 2 numbers 8 and 9 show the Hamming loss results for this feature. Both results were not good enough, where the Hamming loss only increased slightly compared to the Lesk method. Even so, the greater value of concept windows consistently provided an increase in system performance.

5. Conclusion

In this study, we propose word embedding as a feature extraction method with the aim of dealing for weaknesses of Lesk algorithm. Feature extraction is performed on the 3rd order-tensor representation,

where the document is represented by a term-by-concept matrix. We use the Semantic Naive Bayes classification method, which is a development of Naive Bayes to classify data with tensor representation. Based on the experiment results and analysis, classification with features that contain word embedding produce a better performance. This is denoted by the Table 2 number 3–9. However, the best Hamming loss produced in this study was 0.10317 for the prediction of data test.

The stand-alone Lesk feature is poor in providing vector of terms. However, the Lesk feature can improve system performance if combined with word embedding. But not all bundling types perform well. In addition, the concept window on the Lesk parameter can affect system performance. Finally, the best combination of features in this study is the average of Lesk terms vector concatenate with word embedding of term. The future work is to use deep learning classification method combining with our proposed feature extraction.

References

[1] M. P. Akhter, Z. Jiangbin, I. R. Naqvi, M. Abdelmajeed, A. Mehmood, and M. T. Sadiq, "Document-level text classification using single-layer multisize filters convolutional neural network," *IEEE Access*, vol. 8, pp. 42689-42707, 2020. https://doi.org/10.1109/ACCESS.2020.2976744

[2] R. A. Pane, M. S. Mubarok, and N. S. Huda, "A multi-lable classification on topics of Quranic verses in English translation using multinomial naive Bayes," in *Proceedings of 2018 6th International Conference on Information and Communication Technology (ICoICT)*, Bandung, Indonesia, 2018, pp. 481-484. https://doi.org/10.1109/ICoICT.2018.8528777

[3] H. J. Kim, J. Kim, J. Kim, and P. Lim, "Towards perfect text classification with Wikipedia-based semantic Naïve Bayes learning," *Neurocomputing*, vol. 315, pp. 128-134, 2018. https://doi.org/10.1016/j.neucom.2018.07.002

[4] M. N. Al-Kabi, H. A. Wahsheh, I. M. Alsmadi, and A. M. A. Al-Akhras, "Extended topical classification of hadith Arabic text," *International Journal of Islamic Applications in Computer Science and Technology*, vol. 3, no. 3, pp. 13-24, 2015.

[5] G. Mediamer, S. Al Faraby, and Adiwijaya, "Development of rule-based feature extraction in multi-label text classification," *International Journal on Advanced Science, Engineering and Information Technology*, vol. 9, no. 4, pp. 1460-1465, 2019. https://doi.org/10.18517/ijaseit.9.4.8894

[6] M. Y. A. Bakar and S. Al Faraby, "Multi-label topic classification of hadith of Bukhari (Indonesian language translation) using information gain and backpropagation neural network," in *Proceedings of 2018 International Conference on Asian Language Processing (IALP)*, Bandung, Indonesia, 2018, pp. 344-350. https://doi.org/10.1109/IALP.2018.8629263

[7] A. Ta'a, Q. A. Abed, and M. Ahmad, "Al-Quran ontology based on knowledge themes," *Journal of Fundamental and Applied Sciences*, vol. 9, no. 5S, pp. 800-817, 2017. https://doi.org/10.4314/jfas.v9i5s.57

[8] Pew Research Center, "The future of world religions: population growth projections, 2010-2050," 2015 [Online]. Available: http://www.pewforum.org/2015/04/02/religious-projections-2010-2050/.

[9] A. M. K. Izzaty, M. S. Mubarok, N. S. Huda, and Adiwijaya, "A multi-label classification on topics of Quranic verses in English translation using tree augmented Naïve Bayes," in *Proceedings of 2018 6th International Conference on Information and Communication Technology (ICoICT)*, Bandung, Indonesia, 2018, pp. 103-106. https://doi.org/10.1109/ICoICT.2018.8528802

[10] G. I. Ulumudin, A. Adiwijaya, and M. S. Mubarok, "A multilabel classification on topics of qur'anic verses in English translation using k-nearest neighbor method with weighted TF-IDF," *Journal of Physics: Conference Series*, vol. 1192, no. 1, article no. 012026, 2019. https://doi.org/10.1088/1742-6596/1192/1/012026

[11] N. S. Huda, M. S. Mubarok, and Adiwijaya, "A multi-label classification on topics of Quranic verses (English translation) using backpropagation neural network with stochastic gradient descent and Adam optimizer," in *Proceedings of 2019 7th International Conference on Information and Communication Technology (ICoICT)*, Kuala Lumpur, Malaysia, 2019, pp. 1-5. https://doi.org/10.1109/ICoICT.2019.8835362

[12] F. S. Nurfikri, "A comparison of Neural Network and SVM on the multi-label classification of Quran verses topic in English translation," *Journal of Physics: Conference Series*, vol. 1192, no. 1, article no. 012030, 2019. https://doi.org/10.1088/1742-6596/1192/1/012030

[13] M. Biniz, R. El Ayachi, and M. Fakir, "Ontology matching using BabelNet dictionary and word sense disambiguation algorithms," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 5, no. 1, pp. 196-205, 2017. http://doi.org/10.11591/ijeecs.v5.i1.pp196-205

[14] H. J. Kim, J. Kim, and J. Kim, "Semantic text classification with tensor space model-based naïve Bayes," in *Proceedings of 2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Budapest, Hungary, 2016, pp. 4206-4210. https://doi.org/10.1109/SMC.2016.7844892

[15] P. C. Sen, M. Hajra, and M. Ghosh, "Supervised classification algorithms in machine learning: a survey and review," in *Emerging Technology in Modelling and Graphics*. Singapore: Springer, 2020, pp. 99-111. https://doi.org/10.1007/978-981-13-7403-6_11

[16] K. J. Hong and H. J. Kim, "A semantic search technique with Wikipedia-based text representation model," in *Proceedings of 2016 International Conference on Big Data and Smart Computing (BigComp)*, Hong Kong, China, 2016, pp. 177-182. https://doi.org/10.1109/BIGCOMP.2016.7425818

[17] G. Drakopoulos, A. Kanavos, I. Karydis, S. Sioutas, and A. G. Vrahatis, "Tensor-based semantically-aware topic clustering of biomedical documents," *Computation*, vol. 5, no. 3, article no. 34, 2017. https://doi.org/10.3390/computation5030034

[18] L. Zhang, P. Zhang, X. Ma, S. Gu, Z. Su, and D. Song, "A generalized language model in tensor space," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 1, pp. 7450-7458, 2019. https://doi.org/10.1609/aaai.v33i01.33017450

[19] J. Lilleberg, Y. Zhu, and Y. Zhang, "Support vector machines and word2vec for text classification with semantic features," in *Proceedings of 2015 IEEE 14th International Conference on Cognitive Informatics & Cognitive Computing (ICCI*CC)*, Beijing, China, 2015, pp. 136-140. https://doi.org/10.1109/ICCI-CC.2015.7259377

[20] D. Rahmawati and M. L. Khodra, "Word2vec semantic representation in multilabel classification for Indonesian news article," in *Proceedings of 2016 International Conference on Advanced Informatics: Concepts, Theory and Application (ICAICTA)*, Penang, Malaysia, 2016, pp. 1-6. https://doi.org/10.1109/ICAICTA.2016.7803115

[21] D. Rahmawati and M. L. Khodra, "Automatic multilabel classification for Indonesian news articles," in *Proceedings of 2015 2nd International Conference on Advanced Informatics: Concepts, Theory and Applications (ICAICTA)*, Chonburi, Thailand, 2015, pp. 1-6. https://doi.org/10.1109/ICAICTA.2015.7335382

[22] K. S. Eo and K. C. Lee, "Investigating opinion mining performance by combining feature selection methods with word embedding and BOW (bag-of-words)," *Journal of Digital Convergence*, vol. 17, no. 2, pp. 163-170, 2019. https://doi.org/10.14400/JDC.2019.17.2.163

[23] S. Quran, *Cordova Al-Quran dan Terjemahan*. Bandung, Indonesia: Syaamil Quran, 2004.

[24] S. M. Kandi, "Language modelling for handling out-of-vocabulary words in natural language processing," Master's thesis, Department of Mathematics, London School of Economics and Political Science, London, UK, 2018. https://doi.org/10.13140/RG.2.2.32252.08329

Gugun Mediamer **https://orcid.org/0000-0001-9986-6514**

He received B.S. and M.S. degrees in School of Computing from Telkom University in 2019 and 2021, respectively. He is interested in the research area of natural language processing and machine learning.

Adiwijaya **https://orcid.org/0000-0002-3518-7587**

He is a professor of mathematics at School of Computing, Telkom University. He is interested in the research area of graph theory and its applications, data science, and information science. He joined Telkom University since 2000 and has become professor since 2016.