Journal of Computer Science

Original Research Paper

An Implementation of Support Vector Machine on the Multi-Label Classification of English-Translated Quranic Verses

[1]Satrio Adi Prabowo, [2]Adiwijaya, [3]Mohamad Syahrul Mubarok,
[4]Said Al Faraby, [5]Muhammad Zidny Naf and [6]Muhammad Yuslan Abu Bakar

[1,2,3,4,6]*School of Computing, Telkom University, Bandung, Indonesia*
[5]*Telkom Institute of Technology Purwokerto, Purwokerto, Indonesia*

Article history
Received: 24-05-2019
Revised: 26-08-2019
Accepted: 09-12-2019

Corresponding Author:
Adiwijaya
School of Computing, Telkom University, Bandung, Indonesia
Email:
adiwijaya@telkomuniversity.ac.id

Abstract: One of the attempts to understand the meaning and content of the Quran, the central religious text of Islam, is the topic classification of Quranic verses. Verse topic classification aims to help the reader, so he can easily and quickly find information or knowledge contained in the Quran. In this paper, we build a classification model for the topics of English- translated Quranic verses using Support Vector Machine (SVM). The problem of classification of topics of Quranic verses is categorized as a multi-label classification problem. Hence, we design an SVM-based classifier to solve the multi-label classification of topics of Quranic verses. We also implement several techniques such as preprocessing, feature extraction, and dimensionality reduction to solve this problem. Then, we use Hamming Loss as a performance measure to evaluate our proposed classifier model. We find that our proposed model yields outstanding results.

Keywords: Hamming Loss, Quranic Verse Classification, Support Vector Mechine, Weiahted TF-IDF

Introduction

Various attempts have been made to understand the meaning and content of the main holy book of Islam, the Quran. One of the attempts to understand the meaning and the content of the Quran is translation and interpretation. Another attempt such as Quranic verse topic classification can help the reader to find information or knowledge inside the Quran. One of Quranic verse topic classification was created by Dr. Muhammad Hasan Al-Himshy in the Tafsir wa Bayan Al-Quran, published by Dar ar-Rasyid in Damascus. Tafsir wa Bayan Al-Quran classifies Quranic verses by topic or theme contained in each Quran verse and used in many Quran publisher such as Syammil Quran. Besides that, some have also created a digital Quran as an attempt to access Quran in a faster and easier manner.

If one is given digital Quran data and access to topic classification provided by an expert, then the implementation of machine learning to classify Quranic verses by topic may help one to better understand the Quran. One of the popular classifier techniques in machine learning is the Support Vector Machine (SVM) (Kowalczyk, 2017). Besides that, SVM can generalize data with high dimensionality, which is good for textual data, which usually has high dimensionality (Joachims, 1998; Aggarwal and Zhai, 2012).

Before we build the system, we found that about 55% of Quranic verses are multi-label in nature, which means that some verses not only can be considered as one topic but can also be considered as two or more topics. Basically, SVM is a binary classifier. SVM can determine whether or not one data is considered as one class. Therefore, we need to modify SVM so it can be used as a multi-label classifier.

In this paper, we build an SVM-based multi-label classification model for topics of English- translated Quranic verses. We used Tafsir wa Bayan Al-Quran to train the model. After that, we used Hamming Loss as a performance measure of our model. Hamming Loss evaluates the fraction of misclassified instance-labels (Zhang and Zhou, 2014).

The remainder of this paper is organized as follows: in section 2, we provide some previous works that are strongly related to our work. In section 3, we present the details of our method including a brief explanation of each process. Then, in section 4, we present our result evaluation and analysis. Finally, in section 5, we conclude this paper.

Science Publications

Related Work

In text categorization, multi-label classification methods have increasingly been applied (Tsoumakas and Katakis, 2007; Katakis *et al*., 2008) provide an overview of multi-label classification in their work. They also provided an introduction to multi-label classification and performed comparative experiments between multi-label classification methods. Meanwhile, *classifier trellis* (CT) has been proposed to overcome the problem in scability limitations on larger datasets when modeling a fully-cascaded chain (Read *et al*., 2015). Bakar and Faraby (2018; Al Faraby *et al*., 2018) classified hadith of Bukhari into multi-label classification such as suggestion, prohibition and information using some of feature selection and classifier. This feature selection and classifier on text classification has been done previously in research (Pratiwi, 2018; Naf'an *et al*., 2019). They concluded that the use of feature selection had a good impact on the classification results.

Some previous works are strongly related to this work. We found two previous works that solved the multi-label classification of topics of English-translated Quranic verses, which also used the same dataset as our study. Both studies used a probabilistic model to solve this problem.

Izzaty *et al*. (2018) solved the multi-label classification problem of topics of English-translated Quranic verses using a Tree Augmented Naive Bayes (TAN) classifier, yielding the best Hamming Loss value of 0.1121. They used a similar preprocessing technique to that of this paper, and then extracted the feature using bag-of-words. They also used Mutual Information as feature selection. They analyzed some parameters such as the Mutual Information Threshold and the influence of the structure of TAN on Hamming Loss.

Pane *et al*. (2018) solved the multi-label classification of English-translated topics of Quranic verses with a best Hamming Loss value of 0.1247 using the Multinomial Naive Bayes classifier. They used a similar preprocessing technique to this paper, and bag-of-words as feature extraction. They analyzed the effect of stemming and the influence of prior probability value on Hamming Loss.

Unfortunately, both studies used a whole dataset to train and test their model. In this paper, we found that about 45% of the dataset consisted of single-labeled data, which means that we have to make sure that our training set for each fold in the k-fold cross validation contains multi-label data.

Methodology

Figure 1 depicts the general overview of our system. Several processes were carried out before training or testing our classifier including preprocessing, feature extraction, and dimensionality reduction. To evaluate the system, we use Hamming Loss which is an evaluation metric on a multi-label system.

Dataset

We used the same dataset as previous works. This dataset contains English-translated Quranic verses by Sahih International. The dataset contains 6236 verses with 15 main topics. The main topics are Arkanul Islam; Iman; Al-Quran; Knowledge and its various fields; Deeds; Da'wah to Allah; Jihad; Mankind and its Community Relation; Akhlak; Regulation of Wealth; Laws; Societies and Nations; Trades and Agricultures; Histories; and Religions.

As mentioned above, we found that the dataset contained 3412 multi-label verses (about 55% of the total verses) and 2824 single-label verses (about 45% of the total verses). In this paper, we ensured that multi-label data was distributed well in each training set.

Preprocessing

We used preprocessing to assess the quality of our data. We hope that this method can help the model achieve better accuracy and efficiency. An overview of our preprocessing step is shown in Fig. 2:

(i) Cleaning: we used the Python Regular Expression to clean our data. We used this method to eliminate symbols and punctuations in the English translation of the Quranic verses

(ii) Case Folding: after we cleaned our data from unnecessary symbols and punctuations, we then convert our data into lowercase format. We do this via Python str.lower

(iii) Tokenization: then we split each Quran verse into its tokens (words). We used word_tokenize from nltk. tokenize

(iv) Stopword Removal: from the token, we eliminated the token (word) that usually has insignificant meaning. In this paper, we used nltk.corpus to eliminate insignificant words in English

(v) Stemming/Lemmatization: finally, we reduced inflection and changed each token (word) to its basic form (using a dictionary). Stemming uses a heuristic or statistical method to cut off a word. Lemmatization uses morphological analysis and usually refers to a dictionary. In this paper, we analyzed the performance of the WordNetLemmatizer, PorterStemmer, and SnowballStemmer from nltk.stem

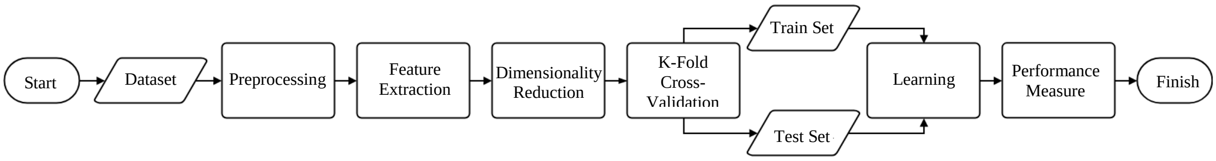

Fig. 1: General overview of our system

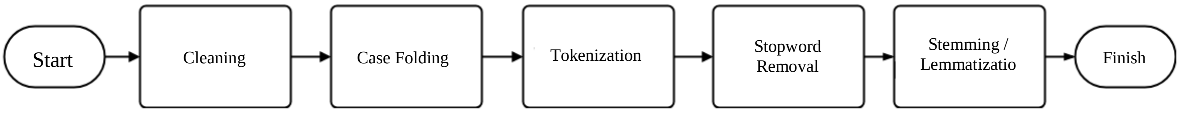

Fig. 2: Preprocessing flow diagram

Feature Extraction

After data preprocessing, we extracted the data feature using a weighted TF-IDF method. Weighted TF-IDF converted our textual data into a geometric space (Han *et al*., 2011). We converted the data into geometric space because we used SVM as our base classifier. SVM is a geometrical model for machine learning (Zhang *et al*., 2017). A geometrical model is a machine learning model that uses a geometric concept such as a point, line, plane, and so on. To convert our data, first, we simply counted each frequency of the word and then normalized it. We then used 1 to compute the weighted Term Frequency (TF):

$$TF(_{a,t}) = \begin{cases} 0 \text{ } if \text{ } freq(_{a,t}) = 0 \\ 1 + \log\left(1 + \log\left(freq(_{a,t})\right)\right) \text{ } otherwise \end{cases} \quad (1)$$

with, *freq* (a, t) is the total number of occurrences of term t in verse *a*. The output of weighted *TF* is a weighted term frequency matrix. Then, we compute the weighted inverse document using Equation 2. The weighted inverse document represents a scaling factor (the important) of term *t*. If term *t* occurs in many documents, its important will be scaled down, vice versa:

$$IDF(t) = \log \frac{1 + |a|}{|a_t|}, \quad (2)$$

where, $|a|$ is the number of all verses and $|a_t|$ is the number of verses that contain term t. The output of Weighted *IDF* is a weighted *IDF* vector. Then, each element on each row of the weighted term frequency matrix is multiplied by the weighted *IDF* vector, denoted by Equation 3:

$$TFIDF(a,t) = TF(a,t) * IDF(t)\,. \quad (3)$$

Dimensionality Reduction

Our weighted *TF-IDF* matrix is a big and sparse matrix. If we used our weighted *TF-IDF* matrix directly with our classifier, it will consume massive amounts of memory and could render operating the big and sparse matrix impossible with such a limited resource. Hence, we used the Truncated Singular Value Decomposition (*SVD*) or known as Latent Semantic Analysis (LSA) (Xu *et al*., 2016) to reduce the dimension of our Weighted *TF-IDF* matrix. Unlike Principle Component Analysis (*PCA*), LSA does not require the data to be centered first. Therefore, we can use LSA directly on big sparse data such as our weighted *TF-IDF* without consuming massive memory. LSA is described by Equation 4:

$$\mathrm{X} \approx \mathrm{X_k} = \mathrm{U_k}\,\Sigma_{\mathrm{k}} \mathrm{V}_k^T, \quad (4)$$

where, U is the SVD term matrix, Σ it the document matrix and k is the number of new features (less than initial number of features).

K-Fold Cross Validation

To evaluate the performance of the classification algorithm in handling our limited data sample, we used k-fold Cross Validation (Flach, 2012; Christopher *et al*., 2008). We split our data into k parts and rotated the training and testing sample. Using this method, we made sure that every data point would be used as training and testing datasets. We also distributed our data to make sure we trained and tested a proportional size of multi-label and single-label data. Then, we computed the average performance using Hamming Loss. We used k-fold cross validation to make sure that our classifier could be generalized.

Support Vector Machine

We used SVM as a base classifier. SVM finds the optimal hyperplane, which is the best separator hyperplane that separates data optimally (Aggarwal and Zhai, 2012). An optimal hyperplane defined by vector *w* and b, such that it will produce the largest margin. To determine the best vector *w* and *b*, we first solve the SVM Optimization Problem defined by Equation 5:

$$\begin{aligned} &\min_{w,b} imize \frac{1}{2} \| w \|^2 \\ &subject\,to\, yi\,(w.xi) + b - 1 \geq 0, i = 1,..,m \end{aligned} \quad (5)$$

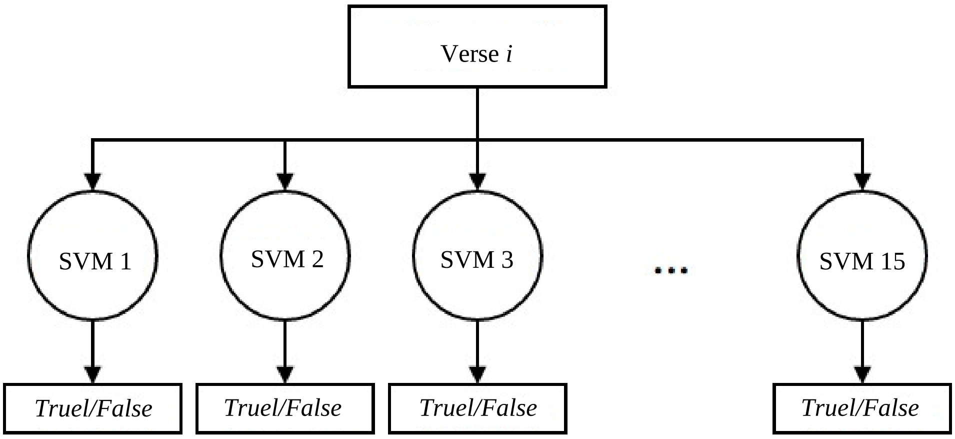

Fig. 3: The implementation of multilabel classification using SVM

In this study, instead of using Quadratic Programming Solver (QP) to solve SVM Optimization Problem, we used a method called Sequential Minimal Optimization (SMO) to deal with high memory consumption. It's because SMO breaks this large QP problem into a series of smallest possible QP problems. These small QP problems are solved analytically, which avoids using a time-consuming numerical QP optimization as an inner loop (Refaeilzadeh *et al*., 2009; Platt, 1998).

Basically, SVM is a binary classifier. Therefore, we have to use a kernel trick to classify non-linearly separable data. The kernel will transform the data into another dimension before it is used to build a model using SVM. We then examine three kernels, which are the Linear kernel, RBF kernel, and Sigmoid kernel. The idea is mapping the non-linear separable data-set into a higher dimensional space where we can find a hyperplane that can separate the samples. This is called kernel trick.

Multilabel Classifier

As mentioned above, SVM is basically a binary classification method. We provide 15 SVMs for each topics of the Quran. Each SVM will determine whether or not the data is classified into topics. This process is shown in Fig. 3.

Performance Measure

We computed the average performance of each k-fold cross validation loop. We used Hamming Loss to compute our model performance. Hamming Loss is defined by Equation 6, as adapted from previous work (Kohavi, 1995):

$$Hamming\,Loss = \frac{1}{NL}\sum\nolimits_{i=1}^{N}\sum\nolimits_{j=1}^{L}\left[\hat{y}_j^{(i)} \neq y_j{}^{i}\right] \qquad (6)$$

where, N is number of data and L is number of labels or classes. If Hamming Loss is zero, this means that the model has classified the data perfectly. Therefore, the smaller the Hamming Loss value, the better the performance of the model.

Evaluation

In this section, we present our results and analysis.

Results

We examined our model using three different SVM kernels: Linear kernel, RBF kernel, and Sigmoid kernel. For each kernel, we examined the number of dimensions and its stem- ming/lemmatization. We used 10, 100, 250, 500, 750, 1000, 1500, 2000, 3000, 4000, 4500, 5000, 5500 and 6000 dimensions to test. For the stemmer/lemmatizer, we examined the Porter Stemmer, Snowball Stemmer and WordNet Lemmatizer. We used 10-fold cross validation for each scenario. We represent our results using a bar diagram.

Linear Kernel Scenario Results

The Linear kernel scenario result is described on Fig. 4. The best Hamming Loss value is 0.09069 using 5000 reduced dimensions and a Snowball Stemmer.

RBF Kernel Scenario Results

The RBF kernel scenario result is described on Fig. 5. The best Hamming Loss value is 0.10652 using 10 reduced dimensions and a Porter Stemmer.

Sigmoid Kernel Scenario Results

The Sigmoid kernel scenario result is described on Fig. 6. The best Hamming Loss value is 0.10651 using 10 reduced dimensions and a Snowball Stemmer.

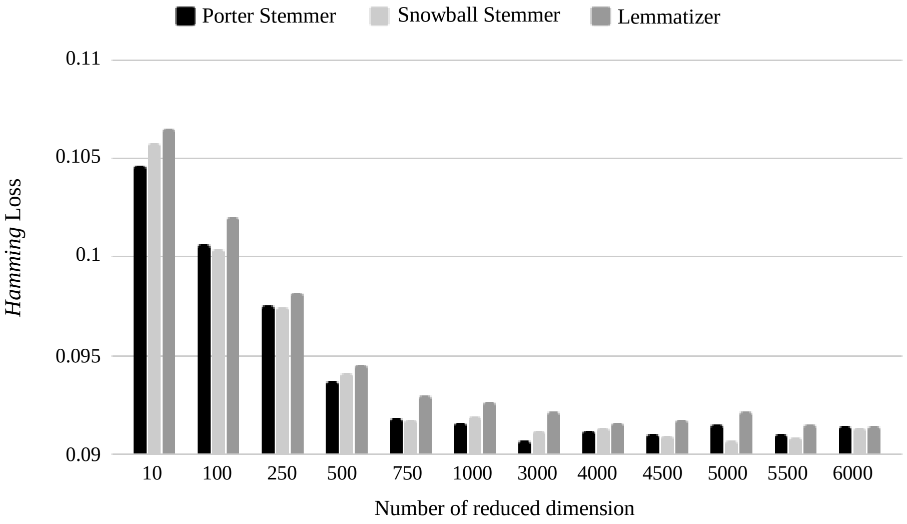

Fig. 4: Linear kernel scenario test results

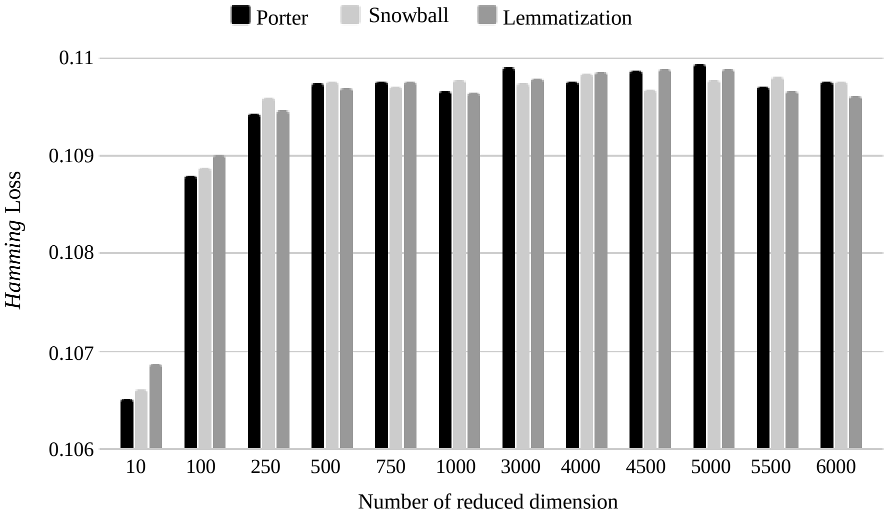

Fig. 5: RBF kernel scenario test results

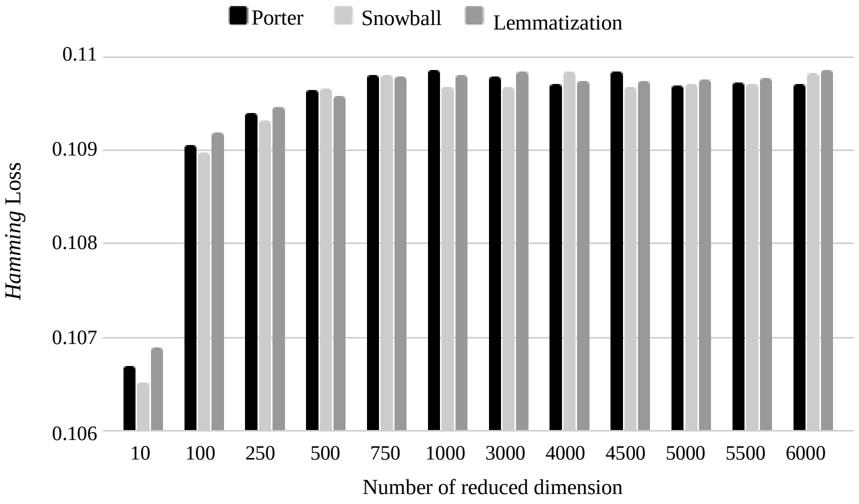

Fig. 6: Sigmoid kernel scenario test results

Analysis Results

From the results above, we conclude that:

(i) Based on the linear kernel scenario results, the Hamming Loss value tends to decrease along with a decrement in reduced dimension. The best Hamming Loss value was achieved in the Linear kernel scenario, where 3000 reduced dimension and a Snowball Stemmer was used, yielding 0.023167
(ii) From the RBF and Sigmoid kernel scenario results, the Hamming Loss value tends to increase along with a decrement in reduced dimension
(iii) There is no significant difference of Hamming Loss value between the Porter Stemmer, Snowball Stemmer or Lemmatizer

Note that this result is sightly higher than the previous work on the same dataset by Izzaty *et al*. (2018) which achieved hamming loss of 0.1121 and Pane *et al*. (2018) which achieved hamming loss of 0.1247. Other works on multi-label clustering but with different datasets have produced hamming losses in range 0.193 on Yeast dataset by Zhang *et al*. (2017) to 0.0118 on bibtex dataset by Xu *et al*. (2016). It's because the use of the kernel and reduced dimension that we do work well on the Quranic data.

Conclusion and Future Work

From the results above, we conclude that the best Hamming Loss value was provided by our multi-label classifier with our dataset consisting of topics of English-translated Quranic verses using SVM. The results was 0.09069 using a Linear kernel, a Snowball Stemmer, and 5000 reduced dimension. Based on our results, in this case, the Linear kernel is the best kernel for this study compared to the RBF kernel and Sigmoid kernel. Also, the different numbers of dimensionality reduction can influence the Hamming Loss value. Besides that, we found that the entire Quranic topics has a complex structure. In future work, the implementation of a classifier using machine learning should be considered for solving the research problem. The analysis of language model and some parameter tuning may also help improve classifier performance.

Acknowledgement

The authors would like to thank the Ministry of Research, Technology and Higher Education, Republic of Indonesia for financially supporting this research.

Author's Contributions

Adiwijaya: Design the research plan (formulation of overarching research goals and aims), validation of algorithm and preparation of article.

Satrio Adi Prabowo: Design and develop of methodology. Verification of the overall of experiments outputs. Conduct a research and investigation process, specifically performing the experiment.

Mohamad Syahrul Mubarok: Implement the computer code, support algorithm and test of existing code components.

Said Al Faraby: Conduct the research and investigation process, specifically performing the experiment and verification of the overall experiment outputs.

Muhammad Zidny Naf: Conduct the research and investigation process, specifically performing the experiment and verification of the overall experiment outputs.

Muhammad Yuslan Abu Bakar: Synthesize study data statistically. Prepare the published work, specifically visualization experiment outputs.

Ethics

This paper is original and has not been published elsewhere. The authors assure that there are no ethical issues that may arise after the publication of thismanuscript.

References

Aggarwal, C.C. and C. Zhai, 2012. Mining Text Data. 1st Edn., Springer Science and Business Media, ISBN-10: 9781461432234, pp: 524.

Al Faraby, S., E.R.R. Jasin and A. Kusumaningrum, 2018. Classification of Hadith into positive suggestion, negative suggestion, and information. J. Phys. Conf. Series, 971: 012046-012046. DOI: 10.1088/1742-6596/971/1/012046

Bakar, M.Y.A. and S. Al Faraby, 2018. Multi-label topic classification of hadith of bukhari (Indonesian Language Translation) using information gain and backpropagation neural network. Proceedings of the International Conference on Asian Language Processing, Nov. 15-17, IEEE Xplore Press, Bandung, Indonesia, pp: 344-350. DOI: 10.1109/IALP.2018.8629263

Christopher, D., P.R. Manning and H. Schtze, 2008. Introduction to information retrieval. Cambridge University Press.

Flach, P., 2012. Machine Learning: The Art and Science of Algorithms that Make Sense of Data. 1st Edn., Cambridge University Press, ISBN-10: 1107096391, pp: 396.

Han, J., J. Pei and M. Kamber, 2011. Data Mining: Concepts and Techniques. 3rd Edn., Elsevier, ISBN-10: 0123814804, pp: 744.

Izzaty, A.M.K, M.S Mubarok, N.S. Huda and K. Adiwijaya, 2018. A multi-label classification on topics of Quranic verses in English translation using tree augmented naive bayes. Proceedings of the International Conference on Information and Communication Technology, May 3-5, IEEE Xplore press, Bandung, Indonesia. DOI: 10.1109/ICoICT.2018.8528802

Joachims, T., 1998. Text categorization with support vector machines: Learning with many relevant features. Proceedings of the 10th European Conference on Machine Learning, Apr. 21-23, Springer, Chemnitz, Germany, pp: 137-142. DOI: 10.1007/BFb0026683

Katakis, I., G. Tsoumakas and I. Vlahavas, 2008. ECML PKDD discovery challenge.

Kohavi, R., 1995. A study of cross-validation and bootstrap for accuracy estimation and model selection. Proceedings of the 14th International Joint Conference on Artificial Intelligence, Aug. 20-25, CA, USA, pp: 1137-1145.

Kowalczyk, A., 2017. Support vector machines succinctly.

Naf'an, M.Z., A.A. Bimantara, A. Larasati, E.M. Risondang and N.A.S. Nugraha, 2019. Sentiment analysis of cyberbullying on instagram user comments. J. Data Sci. Applic., 2: 88-98.

Pane, R.A., M.S. Mubarok, N.S. Huda and A. Adiwijaya, 2018. A multi-label classification on topics of Quranic verses in English translation using multinomial naive bayes. Proceedings of the International Conference on Information and Communication Technology, May 3-5, IEEE Xplore Press, Bandung, Indonesia. DOI: 10.1109/ICoICT.2018.8528777

Platt, J., 1998. Sequential minimal optimization: A fast algorithm for training support vector machines.

Pratiwi, A.I., 2018. On the feature selection and classification based on information gain for document sentiment analysis. Applied Computat. Intelli. Soft Compu. DOI: 10.1155/2018/1407817

Read, J., L. Martino, P.M. Olmos and D. Luengo, 2015. Scalable multi-output label prediction: From classifier chains to classifier trellises. Patt. Recogn., 48: 2096-2109.

Refaeilzadeh, P., L. Tang and H. Liu 2009. Cross-Validation. In: Encyclopedia of Database Systems, Liu, L. and M.T. Özsu (Eds.), Springer, Boston, pp: 532-538.

Tsoumakas, G. and Katakis, 2007. Multi-label classification: An overview. Int. J. Data Warehous. Min., 3: 1-13. DOI: 10.4018/jdwm.2007070101

Xu, C., T. Liu, D. Tao and C. Xu, 2016. Local rademacher complexity for multi-label learning. IEEE Trans. Image Process., 25: 1495-1507.

Zhang, M.L. and Z.H. Zhou, 2014. A review on multi-label learning algorithms. IEEE Trans. Knowl. Data Eng., 26: 1819-1837. DOI: 10.1109/TKDE.2013.39

Zhang, Y., D.W. Gong, X.Y. Sun and Y.N. Guo, 2017. A PSO-based multi-objective multi-label feature selection method in classification. Scientific Rep., 7: 376-376. DOI: 10.1038/s41598-017-00416-0