https://doi.org/10.31436/iiumej.v21i1.1170

STEMMING IMPACT ANALYSIS ON INDONESIAN QURAN TRANSLATION AND THEIR EXEGESIS CLASSIFICATION FOR ONTOLOGY INSTANCES

FANDY SETYO UTOMO[1,2*], NANNA SURYANA[2] AND MOHD SANUSI AZMI[2]

[1]Department of Information System, Faculty of Computer Science,
Universitas AMIKOM Purwokerto, Purwokerto, Indonesia.
[2]Center for Advanced Computing Technology (C-ACT),
Faculty of Information and Communications Technology,
Universiti Teknikal Malaysia Melaka, Melaka, Malaysia.

[*]Corresponding author: fandy_setyo_utomo@amikompurwokerto.ac.id

(Received: 29th May 2019; Accepted: 7th October 2019; Published on-line: 20th January 2020)

ABSTRACT: The current gap that appears in the Quran ontology population domain is stemming impact analysis on Indonesian Quran translation and its exegesis (Tafsir) to develop ontology instances. The existing studies of stemming effect analysis were performed in various languages, datasets, stemming methods, cases, and classifiers. However, there is a lack of literature that studies the stemming influence on instance classification for Quran ontology with different datasets, classifiers, Quran translations, and their exegesis in Indonesian. Based on this problem, our study aims to investigate and analyse the stemming impact on instance classification results using Indonesian Quran translation and their exegesis as datasets with multiple supervised classifiers. Our classification framework consists of text pre-processing, feature extraction, and text classification stage. Sastrawi stemmer was used to perform stemming operation in the text pre-processing stage. Based on our experiment results, it was found that Support Vector Machine (SVM) with Term Frequency-Inverse Document Frequency (TF-IDF) and stemming operation owns the best classification performance, i.e., 70.75% for average accuracy and 71.55% for average precision in Indonesian Quran translation dataset on 20% test data size. While in 30% test data size, SVM and TF-IDF with stemming process own the best classification performance, i.e., 67.30% for average accuracy and 68.10% for average precision in Ministry of Religious Affairs Indonesia dataset. Furthermore, in this study, it was also discovered that the Backpropagation Neural Network has the most precision and accuracy reduction due to the negative impact of stemming operations.

ABSTRAK: Jurang semasa yang muncul dalam domain populasi ontologi Quran adalah punca analisis kesan bendungan pada terjemahan Quran Bahasa Indonesia dan Tafsir bagi membangunkan ontologi kata dasar. Kajian lalu terhadap analisis kesan bendungan telah dijalankan dalam pelbagai bahasa, set data, kaedah bendungan, kes dan pengkelasan. Walau bagaimanapun, terdapat kekurangan kesusasteraan yang mengkaji kesan bendungan dalam bidang pengkelasan ontologi Quran dengan set data berbeza, pengkelasan, penterjemahan Quran dan Tafsir dalam Bahasa Indonesia. Oleh itu, kajian ini bertujuan bagi menyiasat dan menganalisa kesan bendungan terhadap dapatan pengkelasan menggunakan terjemahan Quran dan Tafsir Bahasa Indonesia sebagai set data dengan pelbagai pengkelasan yang diselia. Kaedah pengkelasan kajian ini terdiri daripada pra-pemprosesan teks, ciri pengekstrakan, dan peringkat pengkelasan teks. Pembendung Sastrawi digunapakai bagi menjalankan operasi pembendungan pada peringkat pra-pemprosesan teks. Hasil eksperimen menunjukkan Mesin Vektor Sokongan (SVM), Frekuensi Dokumen Terma Frekuensi-Berbalik (TF-IDF) dan operasi pembendungan memberikan keputusan pengkelasan yang terbaik, iaitu purata ketepatan pada 70.75% dan purata kejituan pada 71.55% terhadap 20% saiz data ujian penterjemahan Quran dalam Bahasa Indonesia. Sementara itu, SVM, TF-IDF dan proses pembendungan memberikan prestasi pengkelasan terbaik, iaitu, purata ketepatan pada 67.30% dan purata kejituan pada 68.10% terhadap 30% saiz data ujian dari set data Kementerian Hal Ehwal Agama Indonesia. Kajian ini juga mendapati Rangkaian Neural Rambatan Belakang menghasilkan pengurangan ketepatan dan kejituan yang paling tinggi disebabkan oleh kesan negatif operasi pembendungan.

KEYWORDS: *K-nearest neighbor; neural network; ontology learning; ontology population; support vector machine*

1. INTRODUCTION

The Quran (Al-Quran) is a Muslim sacred book that contains God’s revelations received by the holy prophet Muhammad (*sallallahu 'alaihi wa sallam*). This holy book contains knowledge, instruction, and scientific facts. Quran consists of several thematic topics or themes such as morals, criminal law, private law, worship, previous nations, the Quran, and faith. These topics aim to guide humankind to reach blessedness in the world and hereafter. The knowledge inside the Holy Quran n could be stored and represented by ontology. There are two approaches to build an ontology, i.e., non-automated and automated process [1]. This automatic process is also known as the ontology population. The non-automated process is usually crafted by a human, such as an ontology engineer or expert in a particular domain, whereas the ontology population is a technique to build an ontology by learning the concepts, relationships, and instances from the text. The standard techniques to conduct the ontology population are lexico-syntactic patterns, classification based on similarity, supervised methods, and knowledge-based and linguistic methods [2].

In the ontology, instances are defined as members of a class [3–5]. Based on earlier research by [6, 7, 8–10], they classified the Quran verses based on thematic topics for the Quran ontology. In their case, thematic topics are concepts or classes, while Quran verses are the instances. Based on their case, our study also adopts the definition of thematic topics as a class and Quran verses as instances. The aim of instances classification is to map the Quran verses into their themes in order for users to have knowledge and better understanding by seeing the entire picture of a particular topic in the Quran.

Stemming is one of the phases in text pre-processing which applies a natural language processing technique for removing affixes from words in order to transform them into their stems [11–13]. The aim of the stemming operation in text classification is to reduce the dimensionality of feature space to provide efficiency within the text classification processing [14,15]. There are several previous studies that employed a stemming operation in the text pre-processing stage to support the instances classification process. Studies by [16–18] performed verse classification for English Quran translation by applying a stemming operation in the pre-processing stage. To classify the verses, research conducted by [16,17] used Back-propagation Neural Network (BPNN) as a classifier, while [18] used three classifiers, i.e., Support Vector Machine (SVM), k-Nearest Neighbour (k-NN), and Naive Bayes (NB). However, their research did not study the stemming impact on classification results. A different approach to performing Quran verse classification was conducted by [19]. In their experiment, they learned the impact of the stemming operation to classify the English Quran verse translations using Hamming Loss as a measuring instrument. As a result, they found stemming was not able to improve Multinomial Naive Bayes performance to classify the instances according to the topics.

To date, the study of stemming impact analysis on instance classification is still a gap that needs to be bridged in the Quran ontology population research field. There is a lack of literature that studies the stemming impact on instance classification with different datasets, Quran translations, and classifiers. Based on this gap, our study aims to investigate and analyse the stemming impact on instance classification results on several datasets and supervised classifiers. Our research contribution is to provide knowledge toward stemming impact on instance classification results in Quran ontology population domain using Indonesian Quran translation and Indonesian Quran exegesis as the dataset.

The rest of the paper is structured as follows: Section 2 presents the study of related work. Section 3 describes our research methodology. Section 4 discusses our experiment results. Finally, we conclude our study results in Section 5.

2. RELATED WORKS

Study to investigate and analyse the stemming impact has been conducted by several previous researchers for some languages and cases. Research by [20] explains the stemming effect on the Arabic text classification. They used *Shereen Khoja's* stemmer and Term Frequency-Inverse Document Frequency (TF-IDF) as a feature selection model. Their dataset consisted of 1100 documents from trusted websites. Then, they classified the entire document info 9 classes, i.e., Agriculture, art, economics, health and medicine, law, politics, religion, science, and sports. Naïve Bayes, SMO (Sequential Minimal Optimization), and Decision Tree (J48) were used as classifiers. The dataset was split 66% for training data and 34% for test data. After two test modes using Percentage Split (PS) and k-fold Cross Validation (CV), it was found that stemming had a negative impact on the classification accuracy of the three classifiers. On PS and CV test mode, J48 had the most significant accuracy decrease from 76.3% to 64.2% in PS mode, and 69.69% to 62.6% in CV mode. Similar research conclusion was also obtained by [21] in their research. They performed Arabic text classification using three datasets that were taken from two trusted sources. The first dataset consisted of 1800 documents with six classes, the second dataset had 1500 documents with five classes, and the third dataset had 1200 documents with four classes. The datasets were split into 70% for training data and 30% for test data on each dataset. Bag of Words (BoW) with sorted and ratio was used for feature selection. They applied the Frequency Ratio Accumulation Method (FRAM) as a classifier, while to transform the words into their root form, they employed Information Science Research Institute's (ISRI) stemmer [22] and *Tashaphyne* stemmer [23]. Experimental results in the entire dataset demonstrated that stemming had a negative impact on the classification accuracy. The most significant accuracy decrease was found in the second dataset from 97.33% to 88.89% with the ISRI stemmer and 95.33% with the *Tashaphyne* stemmer.

Besides Arabic, other studies have studied stemmer impact in another language and dataset. Research by [24] conducted an Indonesian Tweet Classification using 2000 tweets divided into three datasets, i.e., a first dataset with 1500 tweets, second dataset with 1750 tweets, and third dataset with 2000 tweets. They classified the tweets into 2 classes, namely positive and negative tweets. There were 1074 positive and 926 negative tweets. Support Vector Machine (SVM) and Naïve Bayes were used as classifiers. To convert the words into their root form, they applied *Nazief and Adriani's* algorithm. This algorithm is clearly described by [25] in their research. They used BoW and TF-IDF for feature selection in their study. Experimental results on the three datasets demonstrated that stemming had a negative impact on the classification accuracy. The most significant accuracy decrease was seen in the third dataset with BoW as a feature selection and Naïve Bayes as a classifier from 89%

to 85.5%. Furthermore, a study by [19] conducted a multi-label classification on topics of Quranic verses in English translation by Shakir. They used BoW for feature selection, Multinomial Naive Bayes as a classifier, 5-fold cross-validation to evaluate the system, and Hamming Loss as a metric measurement. According to their experiment results, the classification rate without stemming was 0.125 of Hamming Loss, while it was 0.135 using stemming. Based on their research results, it can be concluded that stemming has a negative impact on the classification accuracy.

Different results for English text classification were obtained by [26]. They classified the US Congress data collection document with 60% for training data and 40% for test data. Lovin, Porter, Yet Another Suffix Stripper (YASS), GRAph based Stemmer (GRAS), Statistic Based Stemmer (SNS), and High Precision Stemmer (HPS) were used as stemmers. As a result of text classification by SVM using all stemmers, it was concluded that stemming had a positive impact on the classification accuracy. All stemmers could improve the precision, recall, and f-measure values. The most significant value increase was seen with Porter as a stemmer from 62.1% to 68.3% for precision, 62.9% to 65.3% for recall, and 61% to 65.4% for f-measure.

3. METHODOLOGY

This section is structured as follows: Sub-Section 3.1 discusses the framework for instance classification in this study that was taken from earlier studies. Sub-Section 3.2 provides the collection of datasets used in this investigation. Our experimental configuration is shown in sub-section 3.3. Finally, the experiment test scenario is defined in Sub-Section 3.4.

3.1 Framework Adopted

Based on earlier studies by [19–21, 24, 26], we have adopted their framework for classifying Quranic verses and Quran exegesis instances in our studies. Their framework includes several phases, i.e. text pre-processing, feature extraction, and text classification. Figure 1 presents the instance classification framework in our research. The text pre-processing phase is presented in Sub-Section 3.1.1, whereas Sub-Section 3.1.2 describes the feature extraction and text classification phase.

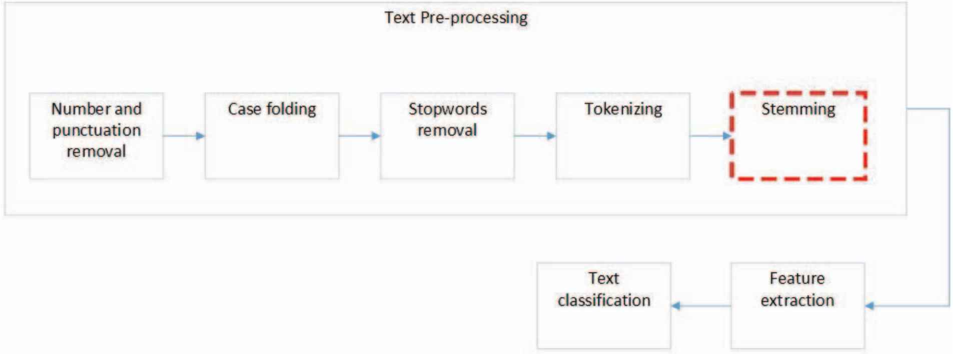


Fig. 1: Instances classification framework.

3.1.1 Text Pre-Processing Phase

The pre-processing phase input is text from Indonesian Quran translation and their exegesis in Indonesian. This phase is aimed at preparing the text in an appropriate format to be processed in the next step. First, it removes the number and punctuation from the text. Then, to prevent ambiguity in term identification, any capital letters discovered are substituted by lower case letters. Common words deemed to have no significance are removed from the sentences in the stop word removal stage after the case-folding procedure. This stage used Tala's stop word list [27] consisting of 757 words. Moreover, in the tokenization phase, the sentence was then divided into words. Tokenization is a process where the text is fragmented into an array of words.

Subsequently, the array of words is used as an input for the stemming phase. Stemming is an operation to remove affixes from the word to convert into their root form. We applied Sastrawi stemmer to perform stemming operation for the Indonesian language text. This stemmer is accessible at *https://pypi.org/project/Sastrawi/*. Sastrawi has work procedures based on the fundamental concept from Nazief and Adriani's stemmer. This stemmer algorithm was described by Asian in [25]. However, there are several modifications on Sastrawi to optimize the stemming operation results. To remove any derivational suffixes, Sastrawi has added the adopted foreign suffix rule {"-is," "-isme," "-isasi"} into the Nazief and Adriani's stemmer origin rule. Furthermore, Sastrawi also has added and modified prefix disambiguation rules to remove complex derivational prefixes {"be-," "te-," "me-," or "pe-"}. Sastrawi stemmer is the optimization result from Nazief and Adriani's algorithm. This stemmer was improved by Confix Stripping (CS) algorithm, Enhanced Confix Stripping (ECS) algorithm, and Modified ECS algorithm [28-29]. Table 1 shows the prefix disambiguation rules that have added and modified in the Sastrawi stemmer.

Table 1: Prefix disambiguation rules

Modified Rules			
Rule	**Construct**	**Return**	**Modified By**
5	beC_1erC_2…	be-C_1erC_2… where C_1!='r'	Sastrawi
12	mempe…	mem-pe…	[25]
14	men{c|d|j|s|t|z}…	men-{c|d|j|s|t|z}…	Sastrawi
16	meng{g|h|q|k}…	meng-{g|h|q|k}…	[25]
17	mengV…	meng-V... | meng-kV... | mengV-... where V='e' | me-ngV…	Sastrawi
18	menyV…	meny-sV… | me-nyV…	[30]
19	mempA…	mem-pA… where A != 'e'	[31]
29	pengC…	peng-C…	[31]
30	pengV…	peng-V... | peng-kV... | (pengV-... if V='e')	[31]
31	penyV…	peny-sV… | pe-nyV…	[30]
Deleted Rules			
Rule	**Construct**	**Return**	**Deleted By**
33	peCerV…	per-erV… where C!={r|w|y|l|m|n}	Sastrawi
New Rules			
Rule	**Construct**	**Return**	**Added By**
35	terC_1erC_2…	ter-C_1erC_2… where C_1 != 'r'	[25]
36	peC_1erC_2…	pe-C_1erC_2… where C_1 != {r|w|y|l|m|n}	[25]
37	CerV…	CerV | CV	[30]
38	CelV…	CelV… | CV…	[30]
39	CemV…	CemV… | CV…	[30]
40	CinV…	CinV… | CV…	[30]
41	kuA…	ku-A…	Sastrawi
42	kauA…	kau-A…	Sastrawi

Based on Table 1, the letter 'C' is a consonant; the letter 'V' is a vowel, and the letter 'A' means any letter. There are 40 prefix disambiguation rules on Sastrawi stemmer, where 32 of these rules were taken directly from Nazief and Adriani's stemmer, and about ten rules were from 32 rules in Nazief and Adriani's stemmer that were modified by several sources.

Sastrawi stemmer has applied the procedure to solve the suffix removal failure that was adopted from [31], for improving the stemming results. This procedure was used to handle the suffix removal problem that arises from the Nazief and Adriani's stemmer. Finally, the array of words that contain the key terms in the basic form is used as an input on the feature extraction stage.

3.1.1 Feature Extraction and Text Classification Phase

Text feature extraction is a method for extracting and selecting text to represent it in a specific form. We used the Bag of Words (BoW) and Term Frequency-Inverse Document Frequency (TF-IDF) model in this research to conduct feature extraction. Bag of Words is an extraction model that could represent text as an unordered set of words and ignore grammatical structure [32]. This model has the representation of a sparse vector that includes appearance for each word in a document. Hereinafter, TF-IDF is a statistical model representing the meaning of a word on a collection by comparing the occurrence of a word in a document with its appearance in another document [33]. Mathematically, the TF-IDF approach can be written in Eq. (1) as follows:

$$TF.IDF = tf_c^r \; x \log(N/df^r) \tag{1}$$

Where tf_c^r is the number of occurrences that term r appeared in a document c, N is the number of entire documents in the corpus, and df^r is the number of documents in which term r appears.

Here is an example to describe the difference between BoW and TF-IDF. Suppose our dataset consists of two Quran verses taken from Indonesian Quran translation. These verses are:

- *Dan jika kamu ditimpa sesuatu godaan syaitan maka berlindunglah kepada Allah* (Surah Al-A'raf: 200)
- *Sesungguhnya syaitan itu tidak ada kekuasaannya atas orang-orang yang beriman dan bertawakkal kepada Tuhannya* (Surah An-Nahl: 99)

After the stop word list and stemming operation are performed on the data set, these are the verses final results after both operations have been conducted.

- *timpa goda syaitan lindung allah* (Surah Al-A'raf: 200)
- *sungguh syaitan kuasa orang orang iman bertawakkal tuhan* (Surah An-Nahl: 99)

The BoW model was built for comparing a set of documents. Based on the data set, the model has two rows, i.e. Surah Al-A'raf: 200 as V_1 and An-Nahl: 99 as V_2.

	timpa	goda	syaitan	lindung	allah	sungguh	kuasa	iman	bertawakkal	tuhan	orang
V₁	1	1	1	1	1						
V₂			1			1	1	1	1	1	2

Based on Eq. (1), the BoW model then modified into a TF-IDF model. The model transformation result is shown below.

	timpa	goda	syaitan	lindung	allah	sungguh	kuasa	iman	bertawakkal	tuhan	orang
V₁	0,3	0,3	0	0,3	0,3						
V₂			0			0,3	0,3	0,3	0,3	0,3	**0,6**

Finally, after feature extraction and selection has been conducted, further the BoW and TF-IDF data are divided into training and test sets. We employed the Back-propagation

Neural Network (BPNN), Support Vector Machine (SVM), and k-Nearest Neighbor (k-NN) classifier to classify the instances. The instances are classified into one of the classes by those classifiers. We utilized three classes, i.e., morals, Al-Quran, and previous nation in our study that taken from thematic topics within Al-Quran Cordoba [34].

3.2 Dataset Collection

We utilized two sources to create the dataset, i.e., the data from *Tanzil* project (*http://tanzil.net*) to build the Indonesian Quran translation and *Quraish Shihab* exegesis corpus; and the data from the Ministry of Religious Affairs Indonesia (*https://quran.kemenag.go.id/*) to develop Quran exegesis corpus. In our study, we utilized several Quran surahs and thematic topics for developing corpus. Table 2 presents the thematic topic number with their names and the total of Quran verse, which is connected to their thematic topic that was used to develop our corpus.

Table 2: Thematic topics and total of Quran verses

Thematic Topic ID	Thematic Topic Name	Total of Quran Verses
1	Morals	218
2	Al-Quran	183
3	Previous Nation	127

Based on Table 2, we employed 528 of Quran verses from Indonesia Quran translation and their exegesis. Table 3 presents the Quran surah, a total of verses inside the surah, and their thematic topic that are utilized to build the corpus.

Table 3: Surah name, total of verses, and their thematic topic

Topic ID	**Al-Baqarah**	**Ali Imran**	**An-Nisa'**	**Al-An'am**	**Al-A'raf**	**At-taubah**	**An-Nahl**	**Taha**
1	51	40	47	13	21	28	14	4
2	59	29	25	20	16	8	18	8
3	13	28	12	10	37	4	5	18
Sum	**123**	**97**	**84**	**43**	**74**	**40**	**37**	**30**

For this study, we utilized Quran verses that are categorized by Al-Quran Cordoba into a single thematic topic. We used two datasets, i.e. Indonesian Quran translation and Quraish Shihab exegesis corpus to observe stemming impact toward the accuracy level of the classifier on both datasets.

3.3 Experimental Setup

We employed two operational frameworks to classify the instances. Figure 2 presents the framework that utilized the BoW approach for classification, while Fig. 3 shows the framework that used TF-IDF to classify the instances.

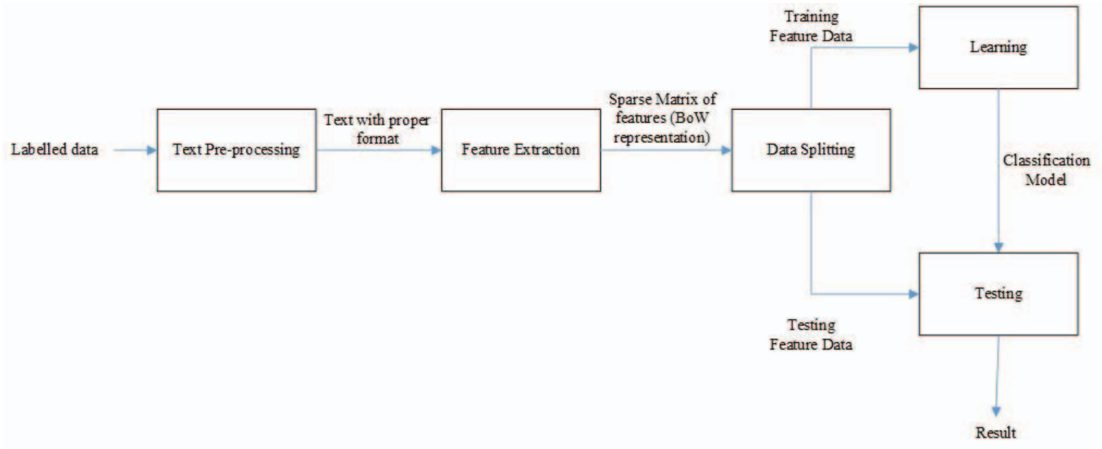


Fig. 2: Framework with BoW approach.

Figure 2 shows that the BoW model follows the feature extraction phase. Furthermore, the data is divided into training and test feature data. The dimensions of both feature data will be presented in Sub-chapter 3.4. For the text pre-processing stage, we used two scenarios, i.e., pre-processing without stemming operation and with the stemming process.

As shown in Fig. 3, BoW model representation is converted into the TF-IDF model. Further, the TF-IDF representation divided into training and test feature data. We developed the operational framework and tested the model performance in Python programming environment. Similar to the previous operational framework, we used two scenarios, i.e., pre-processing without stemming operation and with the stemming process on the text pre-processing stage.

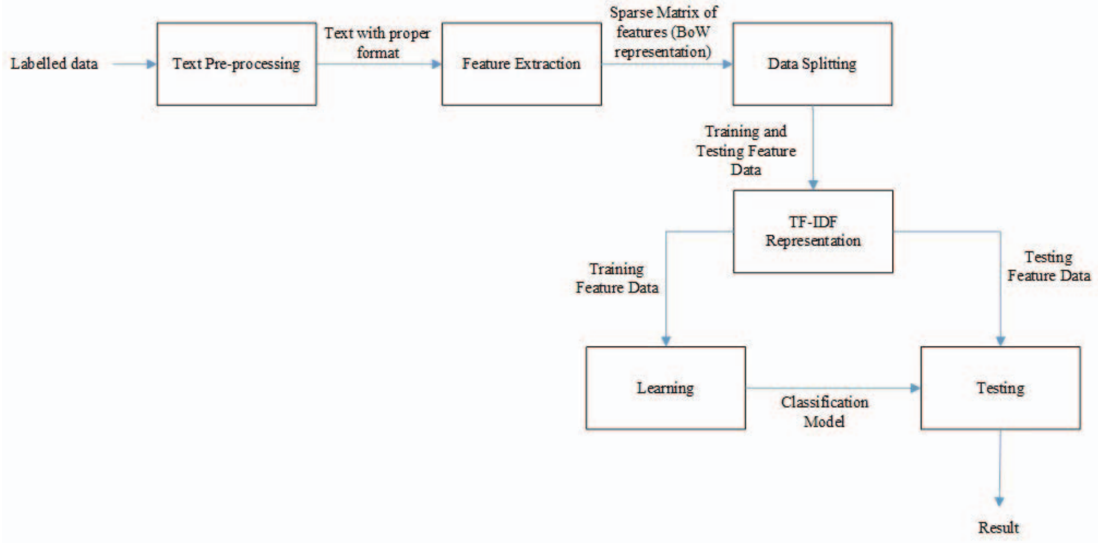


Fig. 3: Framework with TF-IDF approach.

3.4 Test Scenario

We applied several test data sizes to investigate and analyse the stemming operation impact on instance classification with different feature selection models, i.e., BoW and TF-IDF. The test data size for each thematic topic is shown in Table 4.

Table 4: The size of the test data for each topic

Size of test data	Topic 1: Morals (data)	Topic 2: Al-Quran (data)	Topic 3: Previous Nation (data)	Sum (data)
20%	44	37	25	106
30%	66	55	38	159

As shown in Table 4, there are two test scenarios to investigate and analyse the impact of the stemming operation on instance classification performance. We utilized the precision, recall, and accuracy metric for measuring the classification results in this study.

3.5 Metric for Evaluation

In this study, we used various evaluation metrics to measure classification results, i.e. average accuracy, average precision, precision, and recall. Table 5 presents the metric for evaluation and their evaluation focus which is used in this research.

Table 5: The evaluation metric for instance classification

Metrics	**Formula**	**Evaluation Focus**
Precision	$\frac{tp}{tp + fp}$	The ratio between the positive patterns that are correctly predicted from the total predicted data in a positive class
Recall	$\frac{tp}{tp + fn}$	Measure the fraction of positive patterns that are correctly classified
Average Accuracy	$\frac{\sum_{i=1}^{l} \frac{tp_i + tn_i}{tp_i + fn_i + fp_i + tn_i}}{l}$	The average effectiveness of all class from a classifier
Average Precision	$\frac{\sum_{i=1}^{l} \frac{tp_i}{tp_i + fp_i}}{l}$	The average of all class precision

where *tp* is true positive, *fp* is false positive, *fn* is false negative, tp_i are true positive for C_i, tn_i are true negative for C_i, fn_i are false negative for C_i, and fp_i are false positive for C_i.

In our experiment, we used average accuracy and average precision to measure the impact of the stemming operation in all approaches toward classification results for all datasets. While to measure the effect of stemming operation in all methods toward classification results for each theme within all datasets, we used the precision and recall metric.

4. RESULTS AND ANALYSIS

First, we utilized the Indonesian Quran translation (IQT) corpus as a dataset in our experiment. This experiment applied the test scenario based on Table 4. Figure 4 presents the experiment results for the average precision and average accuracy measurement in all approaches.

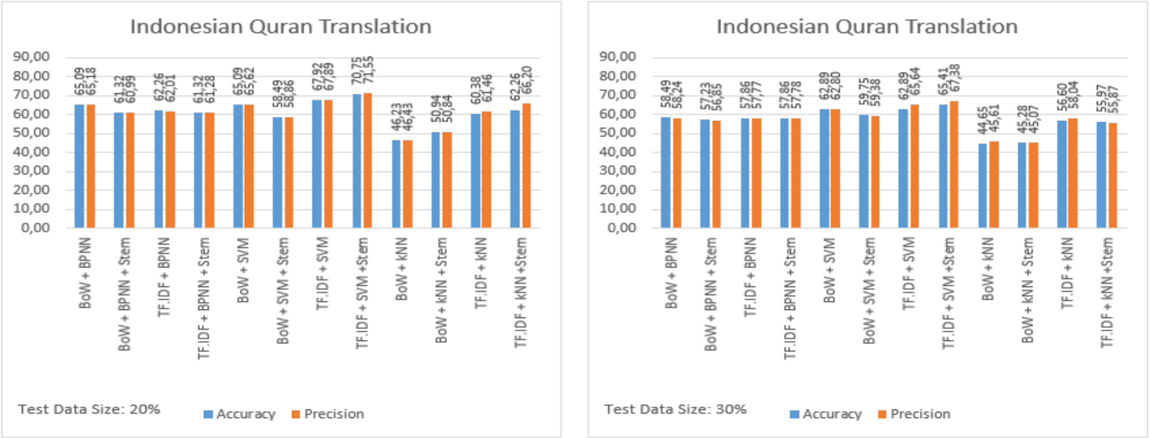


Fig. 4: Measurement average precision and average accuracy on IQT dataset.

As shown in Fig. 4, the stemming operation has a negative impact on BoW/TF-IDF with BPNN approaches for both test data sizes. However, the stemming process has a positive effect on the TF-IDF with SVM approach for both test data sizes. Furthermore, on BoW/TF-IDF with k-NN methods also have a positive impact from the stemming operation on 20% test data size. The TF-IDF with SVM and stemming approach has the highest average precision and average accuracy value on both test data sizes. Hereinafter, Figure 5 shows the experiment results with Quraish Shihab exegesis as a corpus.

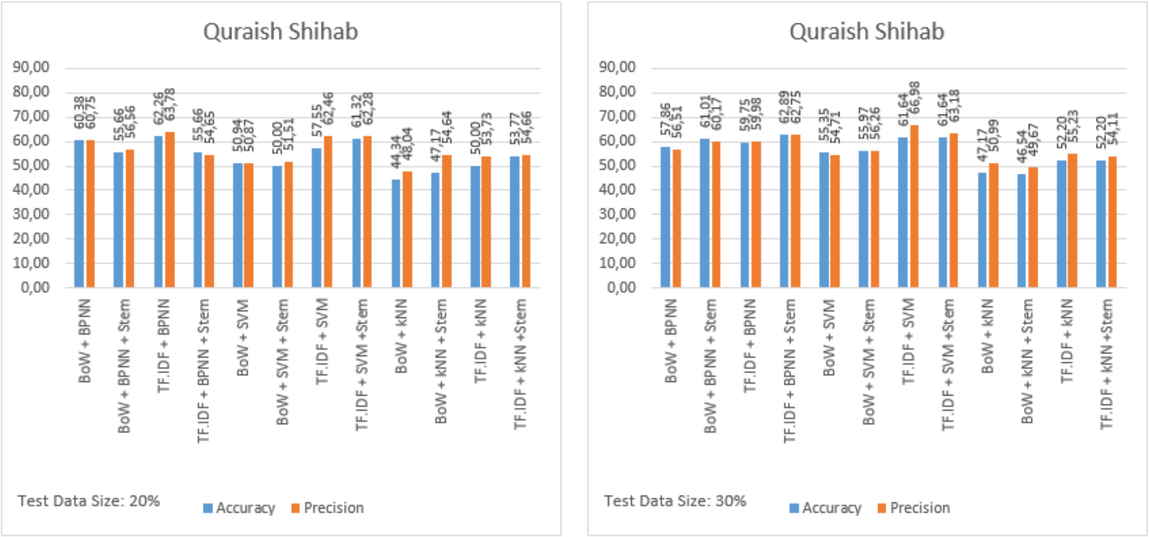


Fig. 5: Measurement average precision and average accuracy on the Quraish Shihab dataset.

Based on Fig. 5, it can be concluded that the stemming operation has a negative impact for BoW/TF-IDF with BPNN approach and BoW with SVM approaches on 20% test data size. Similar to the previous dataset, the BoW/TF-IDF with k-NN methods have a positive impact from stemming operation on 20% test data size. Otherwise, the stemming process has a positive effect for BoW/TF-IDF with BPNN approach and BoW with SVM approaches on 30% test data size. Also, the BoW/TF-IDF with k-NN methods has a negative effect from stemming operation on 30% test data size. Furthermore, Fig. 6 describes the experimental results with the Ministry of Religious Affairs Tafsir as a corpus.

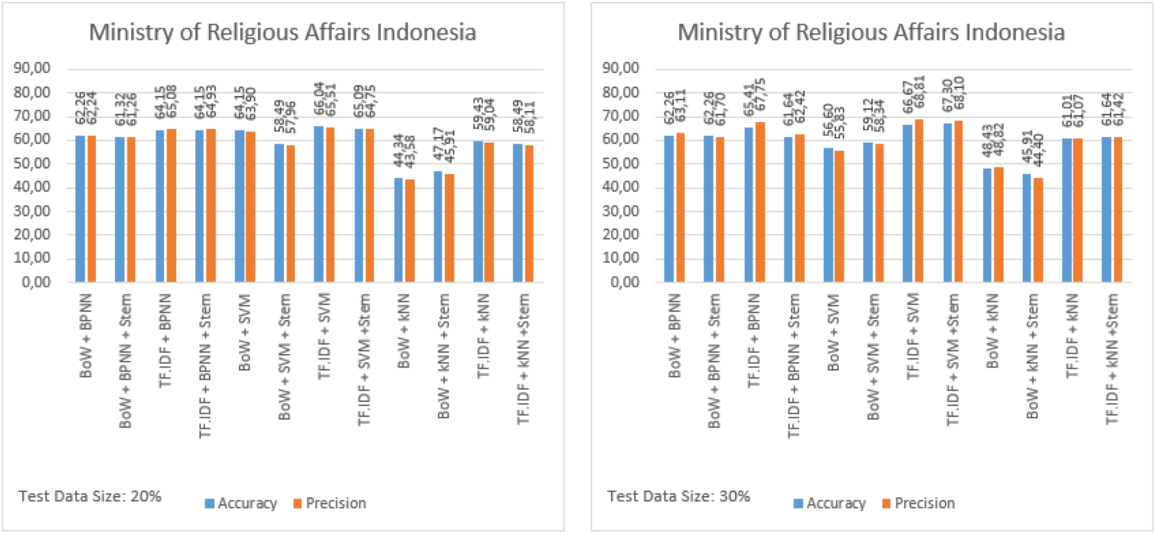
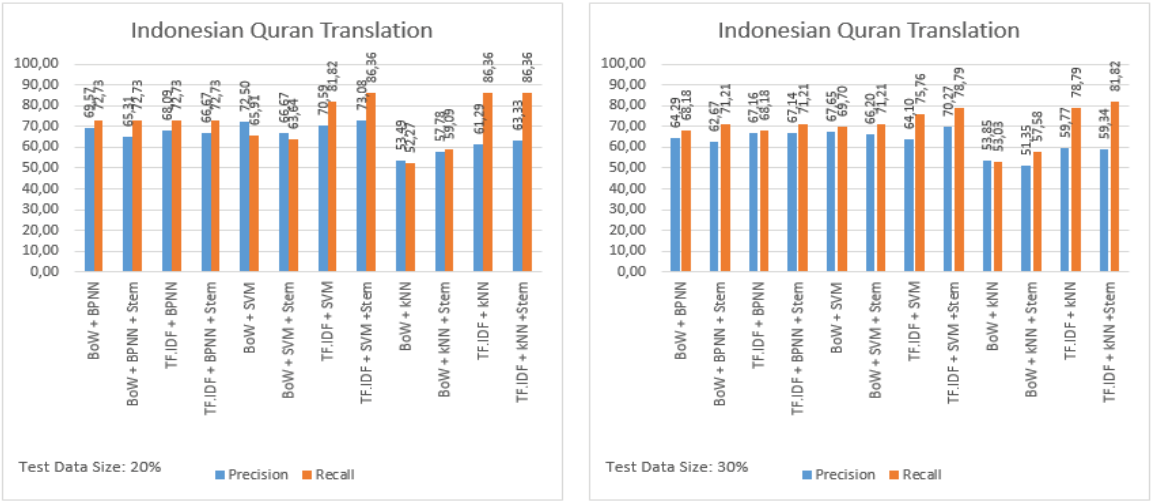


Fig. 6: Measurement of average precision and average accuracy on Ministry of Religious Affairs dataset.

As shown in Fig. 6, it was found that the stemming process has a negative impact on BoW/TF-IDF with BPNN/SVM approaches on 20% test data size. Furthermore, the stemming process also has a negative impact on BoW/TF-IDF with BPNN approaches on 30% test data size. While the BoW/TF-IDF with SVM approaches on 30% test data size has a positive impact from stemming operation.

Figure 7 shows the performance measurement of classification results for Morals class on IQT dataset with 20% and 30% test data size.



(a) (b)

Fig. 7: (a) Morals class on IQT corpus with 20% test data size; (b) Morals class on IQT corpus with 30% test data size.

As shown in Fig. 7, it was found that the stemming operation has provided negative results for BoW/TF-IDF with BPNN and BoW with SVM approaches on both test data sizes since there is a decrease in precision value. However, the stemming process has provided positive results for TF-IDF with SVM and BoW/TF-IDF with k-NN approaches on both test data size since there is an increase in precision and recall value.

Figure 8 presents the performance measurement of classification results for Al-Quran class on IQT dataset with 20% and 30% test data size.

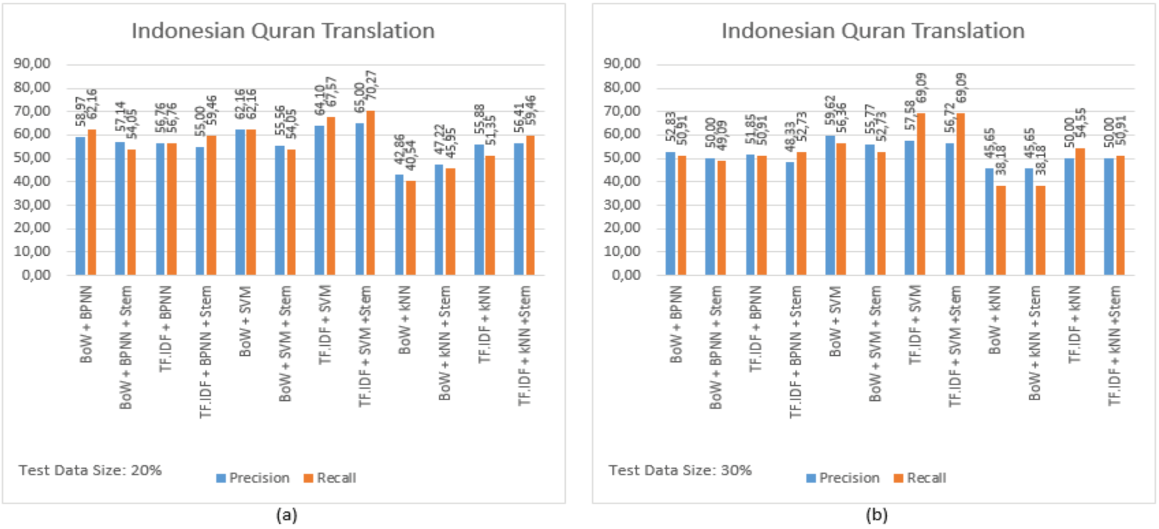


Fig. 8: (a) Al-Quran class on IQT corpus with 20% test data size; (b) Al-Quran class on IQT corpus with 30% test data size.

Based on Fig. 8, similar to the previous class, the stemming process has provided negative results for BoW/TF-IDF with BPNN and BoW with SVM approaches on both test data size since there is a decrease in precision value. While the stemming operation has provided positive results for TF-IDF with SVM and BoW/TF-IDF with k-NN approaches on 20% test data size since there is an increase in precision values.

Figure 9 describes the performance measurement of classification results for previous nation class on IQT dataset with 20% and 30% test data size.

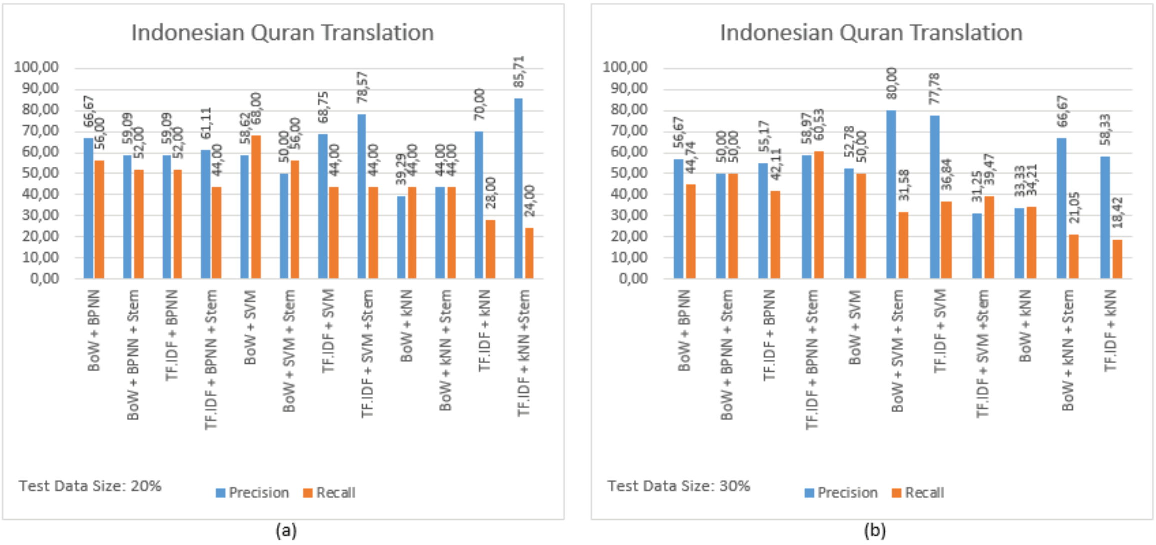


Fig. 9: (a) Previous nation class on IQT corpus with 20% test data size; (b) Previous nation class on IQT corpus with 30% test data size.

According to Fig. 9, it was found that the stemming operation provides a negative impact for BoW/TF-IDF with BPNN and SVM approaches on 20% test data size since there is a decrease in precision and recall values.

Figure 10 describes the performance measurement of classification results for morals class on Quraish Shihab Tafsir dataset, Fig. 11 shows the results for Al-Quran class, while Fig. 12 presents the results for the previous nation class.

As shown in Fig. 10, the stemming process provides a positive impact for BoW/TF-IDF with k-NN and TF-IDF with SVM approaches on 20% test data size since there is an increase in precision and recall values. Furthermore, Fig. 11(a) shows that stemming operation provides a negative impact for BoW/TF-IDF with BPNN and BoW with SVM approaches since there is a decrease in precision values, otherwise Fig. 11(b) presents that the stemming operation provides a positive impact for BoW/TF-IDF with BPNN and BoW with SVM approaches since there is an increase in precision and recall values.

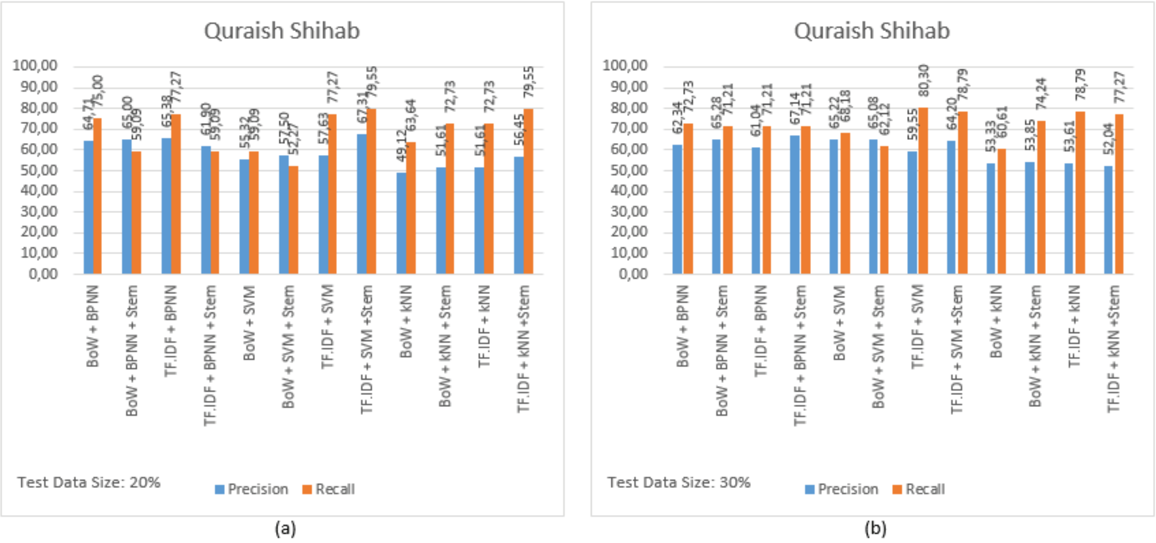


Fig. 10: (a) Morals class on Quraish Shihab Tafsir corpus with 20% test data size; (b) Morals class on Quraish Shihab Tafsir corpus with 30% test data size.

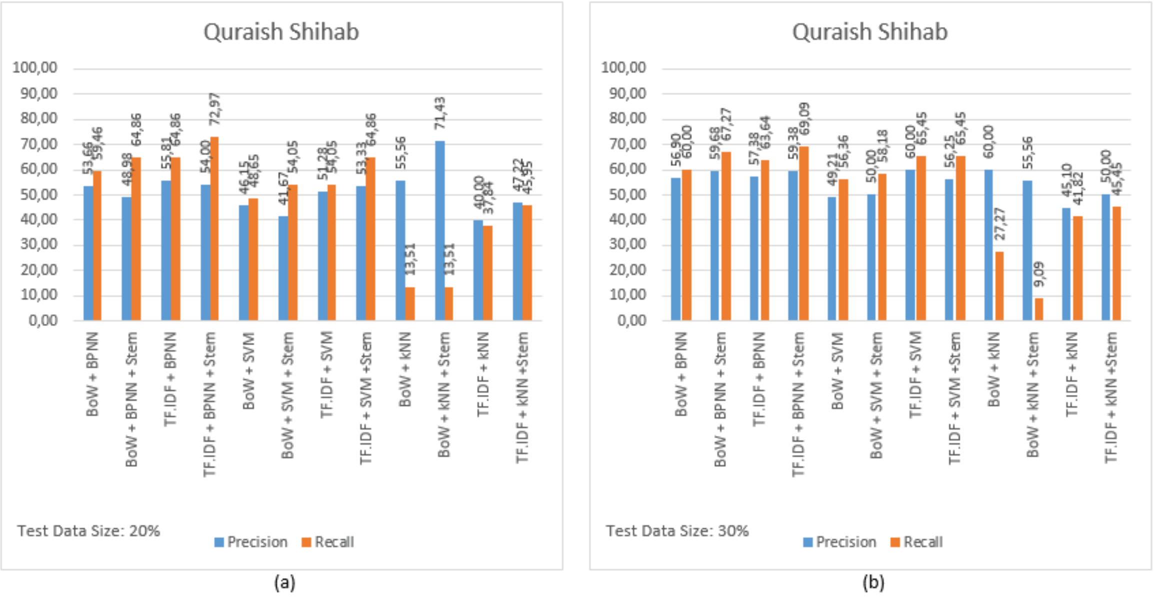


Fig. 11: (a) Al-Quran class on Quraish Shihab Tafsir corpus with 20% test data size; (b) Al-Quran class on Quraish Shihab Tafsir corpus with 30% test data size.

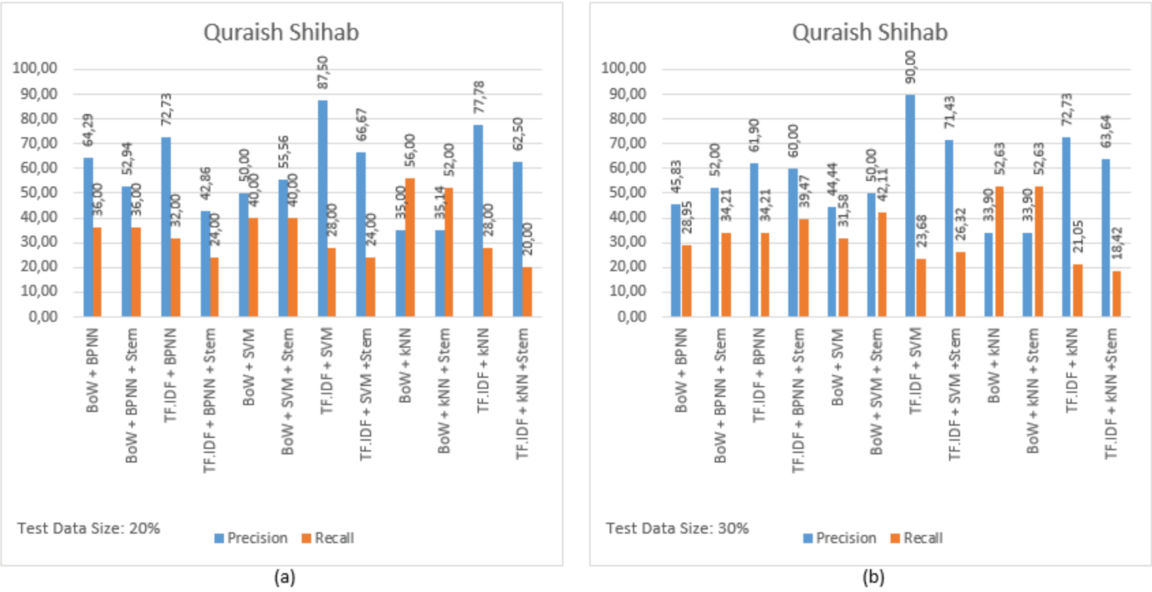


Fig. 12: (a) Previous nation class on Quraish Shihab Tafsir corpus with 20% test data size; (b) Previous nation class on Quraish Shihab Tafsir corpus with 30% test data size.

Figure 13 shows the performance measurement of classification results for morals class on Ministry of Religious Affairs Indonesia dataset, Fig. 14 presents the results for Al-Quran class, while Fig. 15 provides the results for the previous nation class.

As shown in Fig. 13(a), it was found that the stemming operation provides a negative impact for BoW with BPNN/SVM approaches since there is a decrease in precision and recall values. This result is inverse compared to classification results for Al-Quran class with BoW, and BPNN/SVM approaches, as shown in Fig. 14(b). Furthermore, the stemming process also provides a negative impact on all approaches to classify the instances in previous nation class, as shown in Fig. 15(a) and (b).

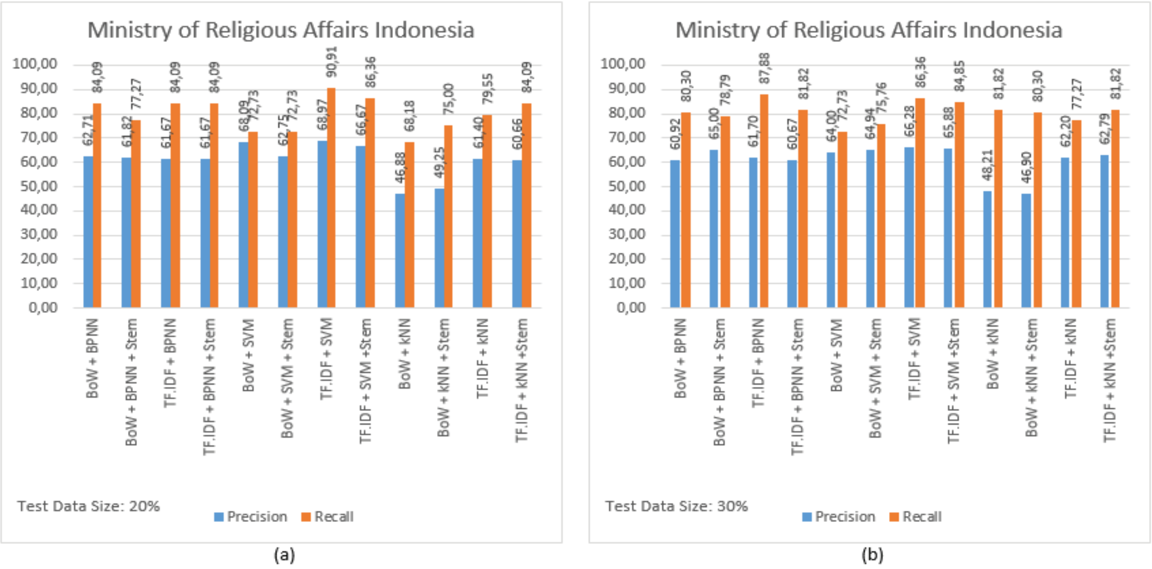


Fig. 13: (a) Morals class on Ministry of Religious Affairs Tafsir corpus with 20% test data size; (b) Morals class on Ministry of Religious Affairs Tafsir corpus with 30% test data size.

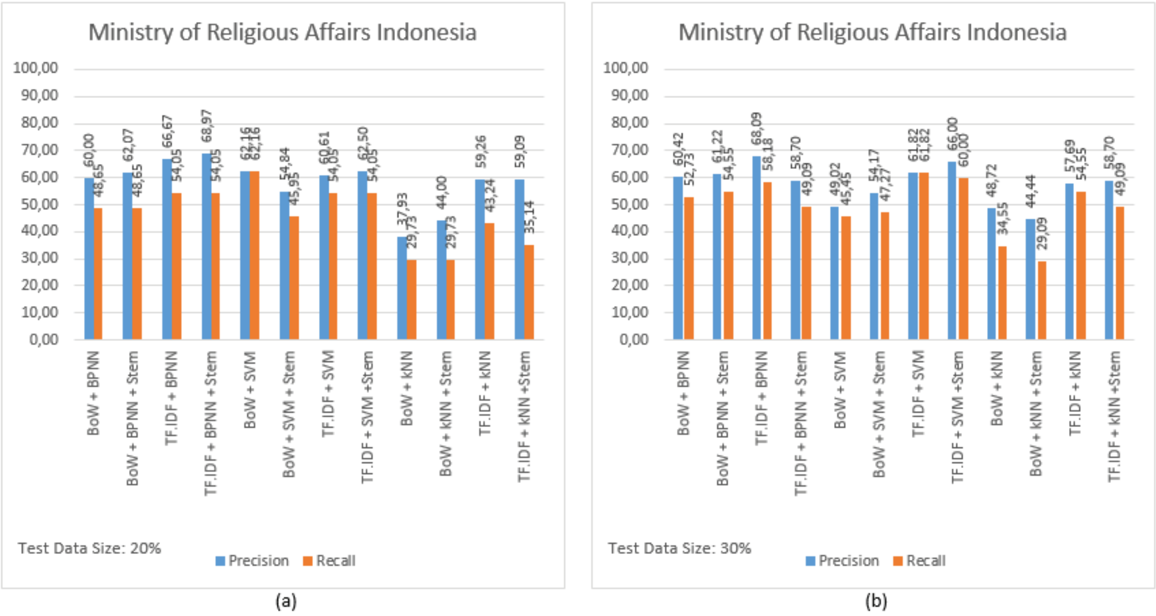


Fig. 14: (a) Al-Quran class on Ministry of Religious Affairs Tafsir corpus with 20% test data size; (b) Al-Quran class on Ministry of Religious Affairs Tafsir corpus with 30% test data size.

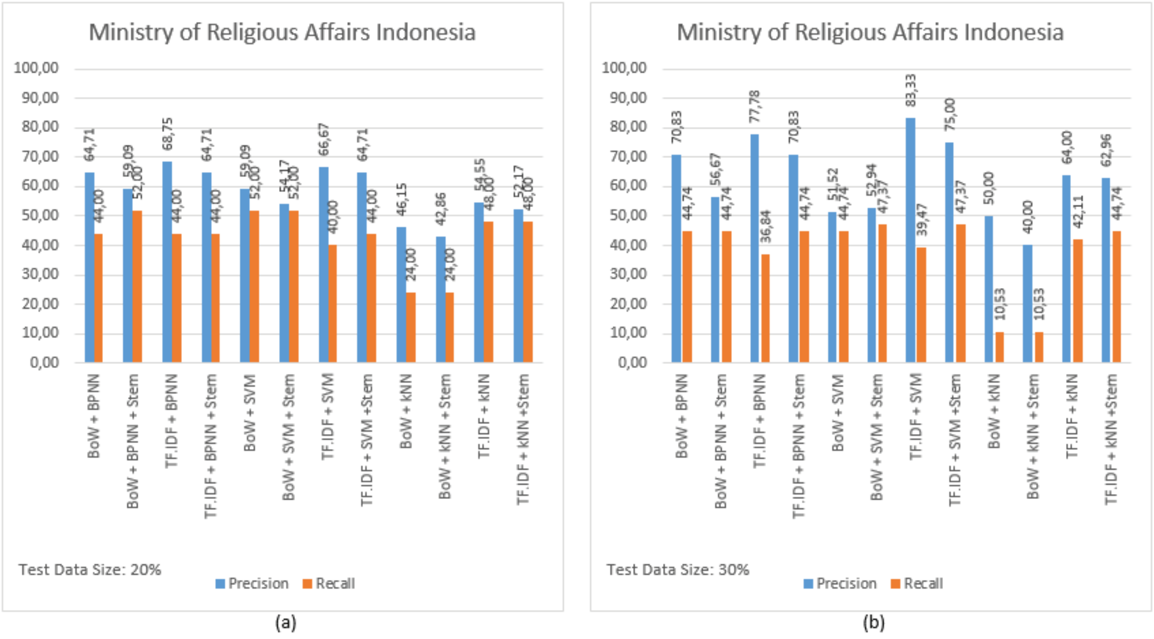


Fig. 15: (a) Previous nation class on Ministry of Religious Affairs Tafsir corpus with 20% test data size; (b) Previous nation class on Ministry of Religious Affairs Tafsir corpus with 30% test data size.

5. CONCLUSIONS

Based on our experimental results, as shown in Fig. 4 to Fig. 6, it was found that the stemming operation provides positive outcomes for k-NN with BoW approach to perform instance classification on 20% test data size. Furthermore, in this test data size, it was also found that the stemming process supplies a negative influence for instance classification with SVM and BoW, BPNN and BoW, and BPNN with TF-IDF. SVM and TF-IDF with stemming operation own the best classification performance, i.e., 70.75% for average accuracy and 71.55% for average precision in IQT dataset. While in the 30% test data size, it was found that stemming operation provides a negative impact on precision for k-NN with BoW approach to classify the instances. However, the stemming process was able to provide a positive effect on accuracy for instance classification with SVM and TF-IDF. SVM and TF-IDF with stemming process own the best classification performance, i.e., 67.30% for average accuracy and 68.10% for average precision in Ministry of Religious Affairs Indonesia dataset. In this study, it was also discovered that the BPNN has the most average precision and average accuracy reduction due to the negative impact of stemming operations.

ACKNOWLEDGEMENT

The authors would like to thank the financial support from Universitas AMIKOM Purwokerto; and Center for Advanced Computing Technology (C-ACT), Faculty of Information and Communications Technology, Universiti Teknikal Malaysia Melaka (UTeM) for their assistance in this research.

REFERENCES

[1] Utomo FS, Suryana N, Azmi MS. (2019). New Instances Classification Framework on Quran Ontology Applied to Question Answering System. TELKOMNIKA, 17(1): 139–146. http://dx.doi.org/10.12928/telkomnika.v17i1.9794

[2] Cimiano P. (2006). Ontology Learning and Population from Text: Algorithms, Evaluation and Applications. New York, Springer Science & Business Media. doi: 10.1007/978-0-387-39252-3

[3] Xian G, Li J, Kou Y, Luo T, Huang Y. (2018). Construction and Application of Upper Country Ontology Based on OWL and SKOS. In Proceedings of the 2nd International Conference on Computer Science and Application Engineering: 22-24 October 2018; Hohhot. pp 1–6. doi: 10.1145/3207677.3278056

[4] Buranarach M, Supnithi T, Thein YM, Ruangrajitpakorn T, Rattanasawad T, Wongpatikaseree K, Lim AO, Tan Y, Assawamakin A. (2016). OAM: An Ontology Application Management Framework for Simplifying Ontology-Based Semantic Web Application Development. International Journal of Software Engineering and Knowledge Engineering, 26(1): 115–145. doi: 10.1142/s0218194016500066

[5] Mitzias P, Riga M, Kontopoulos E, Stavropoulos TG, Andreadis S, Meditskos G, Kompatsiaris I. (2016). User-Driven Ontology Population from Linked Data Sources. In Communications in Computer and Information Science. Volume 649. Edited by Ngonga Ngomo AC., Křemen P. Prague, Springer; 31–41.

[6] Hakkoum A, Raghay S. Advanced Search in the Qur'an using Semantic modeling. (2015). In IEEE/ACS 12th International Conference of Computer Systems and Applications (AICCSA): 17-20 November 2015; Marrakech. pp. 1–4. doi: 10.1109/AICCSA.2015. 7507259

[7] Periamalai NSHA, Mustapha A, Alqurneh A. (2016). An Ontology for Juz' Amma based on Expert Knowledge. In 7th International Conference on Computer Science and Information Technology (CSIT): 13-14 July 2016; Amman. pp. 1–5. doi: 10.1109/CSIT.2016.7549480

[8] Zailani SAM, Omar NA, Mustapha A, Rahim MHA. (2018). Fasting ontology in pillars of Islam. Indonesian Journal of Electrical Engineering and Computer Science, 12(2): 562–569. doi: 10.11591/ijeecs.v12.i2.pp562-569

[9] Ta'a A, Abed QA, Ali BM, Ahmad M. (2016). Ontology-Based Approach for Knowledge Retrieval in Al-Quran Holy Book. International Journal of Computational Engineering Research (IJCER) Ontology-Based, 6(3): 8–15.

[10] Afifi M, Safee M, Saudi MM, Pitchay SA, Ridzuan F, Basir N, Saadan K, Nabila F. (2018). Hybrid Search Approach for Retrieving Medical and Health Science Knowledge from Quran. International Journal of Engineering & Technology, 7: 69–74. doi: 10.14419/ijet.v7i4.15. 21374

[11] Jabbar A, Iqbal S, Khan MUG, Hussain S. (2018). A survey on Urdu and Urdu like language stemmers and stemming techniques. Artificial Intelligence Review, 49(3): 339–373. doi: 10.1007/s10462-016-9527-1

[12] Jabbar A, Iqbal S, Akhunzada A, Abbas Q. (2018). An improved Urdu stemming algorithm for text mining based on multi-step hybrid approach. Journal of Experimental and Theoretical Artificial Intelligence, 30(5): 703–723. https://doi.org/10.1080/0952813X.2018.1467495

[13] Kassim MN, Jali SHM, Maarof MA, Zainal A. (2019). Towards stemming error reduction for Malay texts. Lecture Notes in Electrical Engineering, 481: 13–23. doi: 10.1007/978-981-13-2622-6_2

[14] Uysal AK, Gunal S. (2014). The impact of preprocessing on text classification. Information Processing and Management, 50(1): 104–112. http://dx.doi.org/10.1016/j.ipm.2013.08.006

[15] Sharma D, Jain S. (2015). Evaluation of Stemming and Stop Word Techniques on Text Classification Problem. International Journal of Scientific Research in Computer Science and Engineering, 3(2): 1–4.

[16] Hamed SK, Ab Aziz MJ. (2018). Classification of Holy Quran Translation using Neural Network Technique. Journal of Engineering and Applied Sciences, 13(12): 4468–4475. doi: 10.3923/jeasci.2018.4468.4475

[17] Hamed SK, Ab Aziz MJ. (2016). A question answering system on Holy Quran translation based on question expansion technique and Neural Network classification. Journal of Computer Science, 12(3): 169–177. doi: 10.3844/jcssp.2016.169.177

[18] Rostam NAP, Malim NHAH. (in press). Text categorisation in Quran and Hadith: Overcoming the interrelation challenges using machine learning and term weighting. Journal of King Saud University - Computer and Information Sciences. https://doi.org/10.1016/j.jksuci.2019.03.007

[19] Pane RA, Mubarok MS, Huda NS, Adiwijaya. (2018). A Multi-lable Classification on Topics of Quranic Verses in English Translation using Multinomial Naive Bayes. In Proceedings of the 6th International Conference on Information and Communication Technology (ICoICT): 3-5 May 2018; Bandung. pp. 481–484. doi: 10.1109/ICoICT.2018.8528777

[20] Wahbeh A, Al-Kabi M, Al-Radaideh Q, Al-Shawakfa E, Alsmadi I. (2011). The Effect of Stemming on Arabic Text Classification. International Journal of Information Retrieval Research, 1(3): 54–70. doi: 10.4018/ijirr.2011070104

[21] Sallam RM, Mousa HM, Hussein M. (2016). Improving Arabic Text Categorization using Normalization and Stemming Techniques. International Journal of Computer Applications, 135(2): 38–43. doi: 10.5120/ijca2016908328

[22] Taghva K, Elkhoury R, Coombs J. (2005). Arabic Stemming without a Root Dictionary. In Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC): 4-6 April 2005; Las Vegas. pp. 152–157. doi: 10.1109/ITCC.2005.90

[23] Tashaphyne: Arabic Light Stemmer [https://pypi.org/project/Tashaphyne/].

[24] Hidayatullah AF, Ratnasari CI, Wisnugroho S. (2016). Analysis of Stemming Influence on Indonesian Tweet Classification. TELKOMNIKA, 14(2): 665–673. doi: 10.12928/ telkomnika.v14i2.3113

[25] Asian J. (2007). Effective Techniques for Indonesian Text Retrieval. PhD thesis. RMIT University, School of Computer Science and Information Technology.

[26] Singh J, Gupta V. (2017). A systematic review of text stemming techniques. Artificial Intelligence Review, 48(2): 157–217. doi: 10.1007/s10462-016-9498-2

[27] Tala FZ. (2003). A Study of Stemming Effect on Information Retrieval in Bahasa Indonesia. Master Thesis. Universiteit van Amsterdam, Institute for Logic, Language and Computation.

[28] Kusumaningrum R, Adhy S, Suryono S. (2018). WCLOUDVIZ: Word Cloud Visualization of Indonesian News Articles Classification based on Latent Dirichlet Allocation. TELKOMNIKA, 16(4): 1752–1759. doi: 10.12928/telkomnika.v16i4.8194

[29] Darmawiguna IGM, Pradnyana GA, Santyadiputra GS. (2019). The Development of Integrated Bali Tourism Information Portal using Web Scrapping and Clustering Methods. Journal of Physics: Conference Series, 1165(1): 1–10. doi: 10.1088/1742-6596/1165/1/ 012010

[30] Tahitoe AD, Purwitasari D. (2010). Implementasi modifikasi enhanced confix stripping stemmer untuk bahasa indonesia dengan metode corpus based stemming. Undergraduate Thesis. Institut Teknologi Sepuluh November (ITS), Informatics Engineering Department.

[31] Arifin AZ, Mahendra IPAK, Ciptaningtyas HT. (2009). Enhanced confix stripping stemmer and ants algorithm for classifying news document in indonesian language. Proceeding of International Conference on Information & Communication Technology and Systems (ICTS). pp. 149-158.

[32] Schneider MJ, Gupta S. (2016). Forecasting sales of new and existing products using consumer reviews: A random projections approach. International Journal of Forecasting, 32(2): 243–256. doi: 10.1016/j.ijforecast.2015.08.005

[33] Chen G, Xiao L. (2016). Selecting publication keywords for domain analysis in bibliometrics: A comparison of three methods. Journal of Informetrics, 10(1): 212–223. http://dx.doi.org/10.1016/j.joi.2016.01.006

[34] Kurnia I, Sopian T, Suryana Y, Makbul, Nugraha S, Al-Ghifari, MM, Abdullah R. (2012). Al-Qur'an CORDOBA, 1st ed. Bandung, Cordoba: Internasional - Indonesia.