PAPER NAME

Multi-label Classification.pdf

AUTHOR

Ahmad Rofiqul Muslikh

WORD COUNT

6514 Words

CHARACTER COUNT

34918 Characters

PAGE COUNT

14 Pages

FILE SIZE

520.8KB

SUBMISSION DATE

Feb 22, 2024 11:17 AM GMT+7

REPORT DATE

Feb 22, 2024 11:35 AM GMT+7

● **20% Overall Similarity**

The combined total of all matches, including overlapping sources, for each database.

- 15% Internet database
- Crossref database

- 11% Publications database
- Crossref Posted Content database

● **Excluded from Similarity Report**

- Submitted Works database

- Bibliographic material

Multi-label Classification of Indonesian Al-Quran Translation based CNN, BiLSTM, and FastText

Ahmad Rofiqul Muslikh*[1], Ismail Akbar[2], De Rosal Ignatius Moses Setiadi[3], Hussain Md Mehedul Islam[4]

[1,2]Faculty of Information Technology, Universitas Merdeka, Malang, East Java, Indonesia
[3]Faculty of Computer Science, Universitas Dian Nuswantoro, Semarang, Central Java, Indonesia
[4]Software Engineer, The Mathworks, Inc., United States
Email: [1]rofickachmad@unmer.ac.id, [2]ismaelakbar12@gmail.com, [3]moses@dsn.dinus.ac.id, [4]mehadi.cuet@gmail.com

Abstract

Studying the Qur'an is a pivotal act of worship in Islam, which necessitates a structured understanding of its verses to facilitate learning and referencing. Reflecting this complexity, each Quranic verse is rich with unique thematic elements and can be classified into a range of distinct categories. This study explores the enhancement of a multi-label classification model through the integration of FastText. Employing a CNN+Bi-LSTM architecture, the research undertakes the classification of Quranic translations across categories such as *Tauhid*, *Ibadah*, *Akhlak*, and *Sejarah*. Based on model evaluation using F1-Score, it shows significant differences between the CNN+Bi-LSTM model without FastText, with the highest result being 68.70% in the 80:20 testing configuration. Conversely, the CNN+Bi-LSTM+FastText model, combining embedding size and epoch parameters, achieves a result of 73.30% with an embedding size of 200, epoch of 100, and a 90:10 testing configuration. These findings underscore the significant impact of FastText on model optimization, with an enhancement margin of 4.6% over the base model.

Keywords: Bi-LSTM, CNN, FastText, Multi-label text classification, Quran translation

1. INTRODUCTION

The Qur'an contains important teachings that enable Muslims to follow and adhere to Islamic principles correctly[1]. Categorizing the verses of the Qur'an is an important aspect of its study[2]. This classification concept, as explained by Hasyim, is identified in research by M. Fauzan et al. to cover various themes such as Tauhid, Worship, Muamalah, ethics, Tarikh (History), and Sharia in the teachings of the Qur'an [3]. This makes it easier for people to find and understand the specific verses they need in the Qur'an. Categorizing the verses of the Qur'an is very interesting because of the possibility of each verse falling into one or more categories, as well as the many words with the same meaning often found in the Qur'an translations. This configuration exemplifies multi-label classification.

Natural Language Processing (NLP) technology, especially word embedding, is important in the context of multi-label classification of the Qur'an because many words have similar meanings in the translation[4]. The NLP community's interest in multi-label classification is increasing, encouraging the use of various classification techniques with machine learning and deep learning models to facilitate access, and understanding of the message of the Quran. A number of studies use various machine learning models such as Naive Bayes [5]–[7], random forest[8]–[11], K-Nearest Neighbors (KNN) [7], [9], [12], [13] and Support Vector Machine (SVM) [13], [14] for text classification, produced satisfactory results. However, the focus on multi-label text classification is still limited. For example, Gunawan and Santoso successfully applied SVM to classify Indonesian language news documents from CNN with 90% accuracy for

five main news categories[15]. In contrast, KNN has been utilized in multi-label classification, as demonstrated by Afrian Hanif et al., achieving approximately 91.14% classification accuracy with a Hamming loss value of 0.0886[16].

Recent research has shown improvements in the performance of classification models by implementing deep learning techniques, such as CNNs [17] to classify toxic comments on social media with an accuracy of 99.33%. The use of Long Short-Term Memory Recurrent Neural Networks in classifying public report documents also shows promising results, with the highest f-measure value reaching 88.82%[18]. Other efforts to improve model performance include applying word embedding techniques [19]. Such as in [20] news article classification research using LSTM+Word2Vec, which achieved 95.38% accuracy with an average precision, recall and F1-Score value of around 95%.

Before delving into the specific focus of this research, it is essential to consider the implications of accurate multi-label classification of the Qur'an for various stakeholders. Educators can benefit from more targeted instructional materials, scholars can access categorized verses for in-depth analysis, and the broader Muslim community gains a valuable tool for a nuanced understanding of the sacred text. These implications underscore the real-world significance of advancing multi-label classification techniques for Quranic translations.

Building upon the insights from existing literature, this research specifically focuses on three surahs, namely An-Nisa', Al-Maidah, and Al-An'am, totaling 461 verses. The objective is to explore and analyze how word embedding techniques influence the performance of the multi-label classification process in translating these surahs from the Qur'an. The deep learning model employed, specifically CNN+Bi-LSTM+FastText. The aim is to contribute to developing more advanced and precise tools for analyzing religious texts, with a particular focus on the intricate categorization of verses within these selected surahs.

2. RESEARCH METHOD

The research method proposed in this study consists of several stages illustrated in Figure 1. More details of the research stages are presented in sections 2.1 to 2.8.
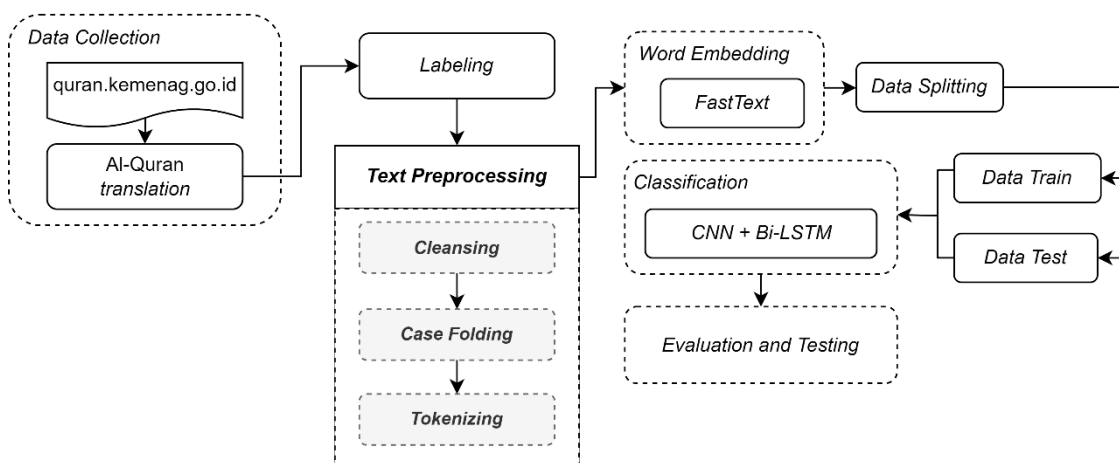

Figure 1. Overview of the proposed Multi-label Classification Pipeline

2.1. Data Collection

To classify multi-label text in this research, an object in the form of text must exist. The data source for this research is the Indonesian translation of the Al-Qur'an, conducted by the Ministry of Religion of the Republic of Indonesia in 2022 and accessible at https://quran.kemenag.go.id. Then, look for the Lajnah Pentashihan Mushaf Al-Qur'an page which is located in the footer of the main page of the website, navigate to the download page and select the "Translate Al-Qur'an" option. You will find a downloadable file with the name "Al-

Quran Translation_v161122.rar". After downloading the file, you can extract it to gain access to the translated text of the Al-Qur'an in Indonesian which will be used as an object in this multi-label text classification research.

2.2. Data Labelling

This research required interpretation and language experts to carry out the labeling manually. The label includes *Tauhid*, which describes the text that states the oneness of Allah Subhanahu Wa Ta'ala [21]. *Ibada*h describes the meaning of submitting and obeying all the prohibitions of Allah Subhanahu Wa Ta'ala [22] and *Akhlak*, which describes a person's character and traits so that they produce good and bad deeds [23]. Finally, *sejarah* is defined as literature that contains stories or events from the past that tell the history of previous nations for the next generation to study [24].

2.3. Text Preprocessing

A text preprocessing step is very crucial and required to make the data set into structured text and ready to be processed in the next step. Additionally, this method has the potential to improve the performance and accuracy of classification models [25]. In this research, the text preprocessing process consists of three stages. The first stage is cleaning, which means deleting or cleaning translation data that contains numbers and delimiters such as commas (,), quotation marks ("), and periods (.). This ensures that the text is free of extraneous elements that could interfere with the analysis. Next, we emply case folding to standardize the text. All characters can be converted into letters of the same type using the case folding method [26]. Next, tokenization is carried out to divide sentences, paragraphs, or documents into parts called tokens [27].

2.4. Word Embedding

Word embedding techniques represent words in a document with real-valued vectors for each word in the vocabulary, ensuring that words with similar meanings have similar representations[28]. This research employs FastText for word embedding, a technique developed by Facebook's research team that leverages character n-grams to create robust vector representations of words [29]. Due to its capability of rapid generation of word embeddings and efficient text classification, FastText has emerged as an important technique for a multitude of Natural Language Processing (NLP) applications.The FastText embedding provides character-level n-grams to each word in the corpus, resulting in richer vector embeddings[30]. It operates on the principles of the Skip-Gram and Continuous Bag of Words (CBOW) models to compute word representations in context. The CBOW model is preferred in configurations where the dataset contains a high frequency of certain words, as it has demonstrated superior performance in capturing the semantic essence of common terms [31]. FastText differentiates itself by assigning vector representations not only to whole words but also to their constituent n-grams, which facilitates the creation of embeddings for words that may not appear in the training corpus. Furthermore, FastText's embedding layer has shown enhanced performance in syntactic tasks, a benefit that becomes particularly pronounced when dealing with smaller corpora [32]. The parameters tested on the FastText word embedding are shown in Table 1.

Table 1. Word Embedding Parameter

Parameter	Values
Embedding Size	100, 200, 300
Epoch	20, 50, 100

Table 1 shows the parameters used for this research, especially in testing fastText word embedding. The dimension of a word vector is determined by the Embedding Size, which can affect the depth of the word representation. Epoch influences how often a word must appear in the corpus to be included in the model.

2.5. Data Split

Training and test data are the two main components of a research data set. Training data

helps machine models understand the patterns and characteristics of the current dataset. After the model is trained with the training data, the test data is used to test the model's performance. This test data is critical for evaluating how well the trained model can generalize well on new data [33]. Table 2 shows the distribution of proportions between training data and test data in research or model development.

Table 2. Splitting Proportion Data

Training Data	Testing Data
60%	40%
70%	30%
80%	20%
90%	10%

Each row in Table 2 shows the combined training and test data percentage used. For example, the first row shows a split where 60% of the entire dataset is used as training data, and the remaining 40% is used as test data. Likewise, for the second row, the training and test data division is 70% and 30% of the entire dataset, respectively. And so on for the third and fourth rows with proportions of 80%:20% and 90%:10% respectively.

2.6. Convolutional Neural Network (CNN)

Convolutional Neural Networks (CNNs) are artificial neural networks that utilize feedforward connections to establish a hierarchical structure in data [34]. Through the learning process, CNN can capture the internal feature representation of data and generalize from these features to new data, especially in the context of object recognition and computer vision problems in images[35]. While commonly used for images processing related applications, CNNs have also been broadly used in solving natural language processing and speech recognition problems [36]. With its flexibility, CNN is not limited to certain data types, so it can be adapted for various complex data analysis applications. The CNN architecture can be seen in Figure 2, which still consists of a combination of convolutional, pooling, and fully connected layers. For input text, first convert it into a sentence matrix $S \in R^{n \times d}$, where n is the number of words in the text, and d denotes the dimension of the word embedding vector for each word, which is analogous to the original pixel points in Figure 2.
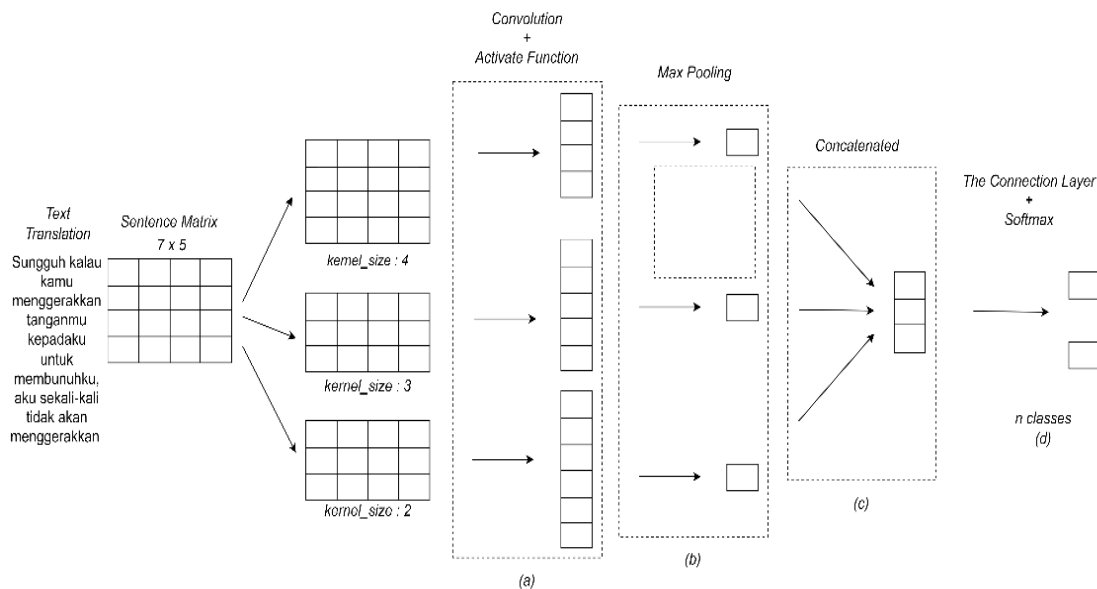

Figure 2. CNN Architecture used.

Subsequently, the sentence matrix undergoes convolution with multiple kernels $W \in \mathbb{R}^{h \times d}$. The ReLU function is applied to the output, with h indicating the kernel size. Texts of varying lengths are converted to fixed lengths, while the feature mapping vectors obtained through convolution are fed into the maximum pooling layer to obtain higher-level features. As illustrated in Figure 2 [37], the input sentence matrix is processed through various filters of different sizes to extract diverse features, and the max-pooling operation follows the corresponding features to obtain the most important features for each dimension. These features are aggregated and forwarded to the Softmax layer within the fully connected layers to output the probability distribution on the labels.
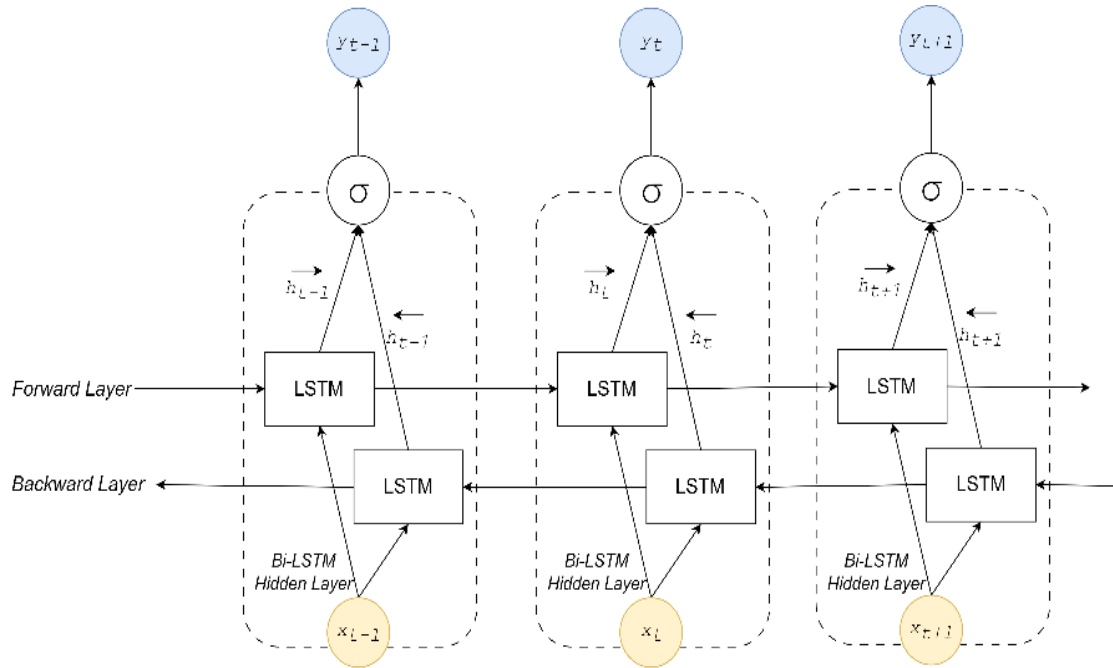

Figure 3. Bi-LSTM architecture used.

2.7. Bi-directional Long Short-Term Memory (BiLSTM)

Bi-LSTM is a neural network architecture consisting of two LSTM layers: forward LSTM to model the previous context and backward LSTM to model the next context[38], [39]. The outputs of these two LSTM layers are concatenated or combined to form the final output. This architecture is particularly useful for sequence processing tasks, such as pattern recognition in sequential data or predicting future values based on both preceding and succeeding contexts[40]. Layers of neurons in Bi-LSTM allow for capturing information from both past and future states simultaneously, enhancing its capability in modeling temporal dependencies. The Bi-LSTM architecture can be seen in Figure 3.

As shown in Figure 3 [41], the Bi-LSTM model extends the standard LSTM by incorporating a second layer that operates in reverse. This model effectively finds text patterns within phrases because each word is evaluated sequentially. The front layer processes the first word to the last word, while the back layer processes the first word to the last word. Therefore, these layers work in two opposite directions so that the model process evolves and allows a better understanding of the context of the text.

2.8. Evaluation

The F1-score of each label measures the accuracy of this research. The F1-score is calculated from the harmonic mean between precision, and recall, and the value ranges between 0 and 1. The F1-score calculation is shown in Equation (1). Precision is the level of accuracy between the information requested by the user and the system answer. Nonetheless, recall is the percentage of

positive items identified correctly. Precision and recall calculations can be seen in Equations (2) and (3).

$$F1 - Score = \frac{2 * Precision * Recall}{Precision * Recall} \tag{1}$$

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

Where in Equations (2) and (3) TP (true positive) is the number of positive predictions with the class target actually being positive, FP (false positive) is the number of positive predictions even though the class target is actually negative, and FN (false negative) is the number of predictions negative even though the class target is actually positive.

3. RESULTS AND DISCUSSION

This research uses translation data from the Al-Qur'an from the Ministry of Religion of the Republic of Indonesia, which consists of 6,236 verses. The research focuses on three surahs, namely An-Nisa', Al-Maidah, and Al-An'am, with a total of 461 verses in the translation of the Al-Qur'an. The selection of three surahs, namely An-Nisa', Al-Maidah, and Al-An'am, in this research may be based on several considerations. First, these surahs were probably chosen because they cover a wide range of themes and teachings that are important in Islam, including social laws, principles of justice, and aspects of everyday Muslim life. Second, there may be a need to focus the analysis on surahs that are relatively shorter compared to the entire Qur'an, thereby simplifying the process of manual labeling and data analysis. The labeling process is carried out manually by experts with four label categories: *Tauhid*, *Ibadah*, *Akhlak*, and *Sejarah*, given a value of 1 if the verse falls into the label category and a value of 0 if not, as in Table 3.

Tabel 3 Labelling Results

Translation	*Tauhid*	*Ibadah*	*Akhlaq*	*Sejarah*
Tiada sekutu bagi-Nya; dan demikian itulah yang...	1	1	1	0
Dan Dialah yang menjadikan kamu penguasa ...	1	0	1	0
Dan janganlah kamu serahkan kepada orang ...	0	0	1	0
Tiada sekutu bagi-Nya; dan demikian itulah yang...	1	1	1	0

The distribution of labels shows that there are 324 verses labeled as *Tauhid*, 205 verses as *Ibadah*, 265 verses as *Akhlak*, and 141 verses as *Sejarah*, as shown in Figure 4. Subsequent to the initial data preparation, the dataset is subjected to a cleansing procedure in which all text characters are transformed to lowercase, and sentences are partitioned into constituent segments. This preprocessing step results in a refined dataset of the Al-Qur'an translation, which is then employed for the training and testing phases. The details of the text preprocessing results are shown in Table 4. The result of the text preprocessing process above is original text that has been modified according to the preprocessing steps described previously. The original text has been converted to all lowercase, numbers have been removed, punctuation characters have been removed, and excessive whitespace has been removed. Thus, the text has been simplified and is ready for use in the next stage.
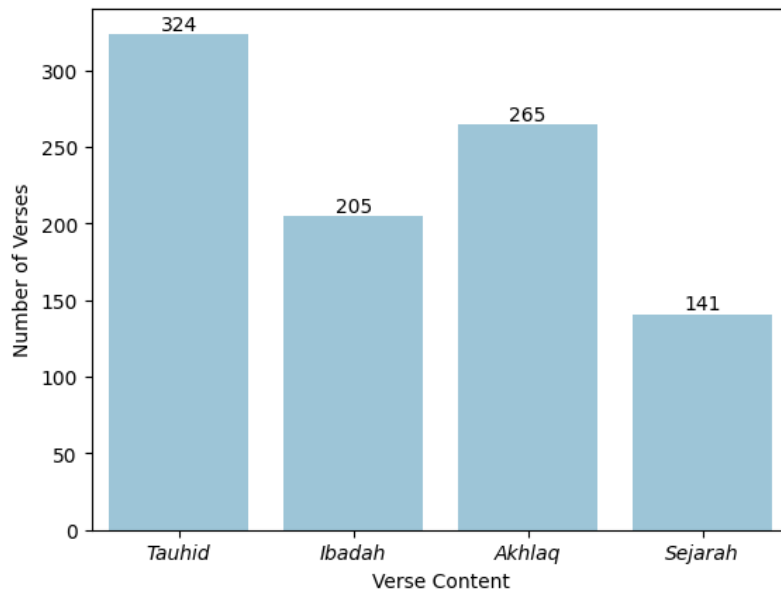
Figure 4. Label Distribution.

Table 4. Sample Text Preprocessing Results.

Original Text	After Preprocessing
Mereka menjawab: "Maha Suci Engkau, tidak ada yang kami ketahui selain dari apa yang telah Engkau ajarkan kepada kami; sesungguhnya Engkaulah Yang Maha Mengetahui lagi Maha Bijaksana".	*Mereka menjawab maha suci engkau tidak ada yang kami ketahui selain dari apa yang telah engkau ajarkan kepada kami sesungguhnya engkaulah yang maha mengetahui lagi maha bijaksana*

3.1. Splitting Data

The split of training and testing data in classifying translations of the Indonesian Al-Qur'an is an important step for accurate model evaluation. Based on Table 2, there are four configurations. In the first configuration, 60% of the training data and 40% of the testing data are allocated, so the total data is 168 for training, 113 for validation, and 180 for testing. In the second configuration, 70% of the training and 30% of the testing data are allocated, resulting in 224 data for training, 96 for validation, and 141 for testing. In the third configuration, 80% of training and 20% of testing data are allocated, resulting in 296 training data, 74 testing, and 141 for testing. In the fourth configuration, 90% of training and 10% of testing data are allocated, resulting in 372 training data, 42 validation, and 47 for testing. Each configuration represents a unique allocation to each stage of the classification model testing.

3.2. CNN + Bi-directional LSTM

After the preprocessing process, CNN + Bi-LSTM testing is carried out at this stage. Before creating a model, the first step is to determine the longest sequence of word tokens the model can accept. This is done by calculating the distribution of word token lengths for each sentence in the data. Figure 6 displays a histogram with a density curve, which depicts the distribution of the number of tokens in a text dataset. This histogram shows the frequency (or density) of tokens on the vertical (Y) axis and the number of tokens on the horizontal (X) axis. The graph shows that most of the data is concentrated around the lower values of the number of tokens, with a density peak around 25 to 50 tokens. This distribution decreases significantly as the number of tokens increases, indicating less text with a higher number of tokens. This is common in text datasets, where most texts are short and only a few are very long. The density curve drawn over the histogram follows the shape of the histogram. It shows an estimate of the continuous distribution of the random variable, which in this case is the number of tokens. The curve indicates that the distribution of the number of tokens approaches a normal distribution for low values but has a

long tail to the right, indicating that the distribution is likely skewed to the right or positively skewed. This suggests that there are a number of texts whose length exceeds the average but not enough to affect the general shape of the distribution as a whole. Therefore, the max_seq_length value for the model used is 150.
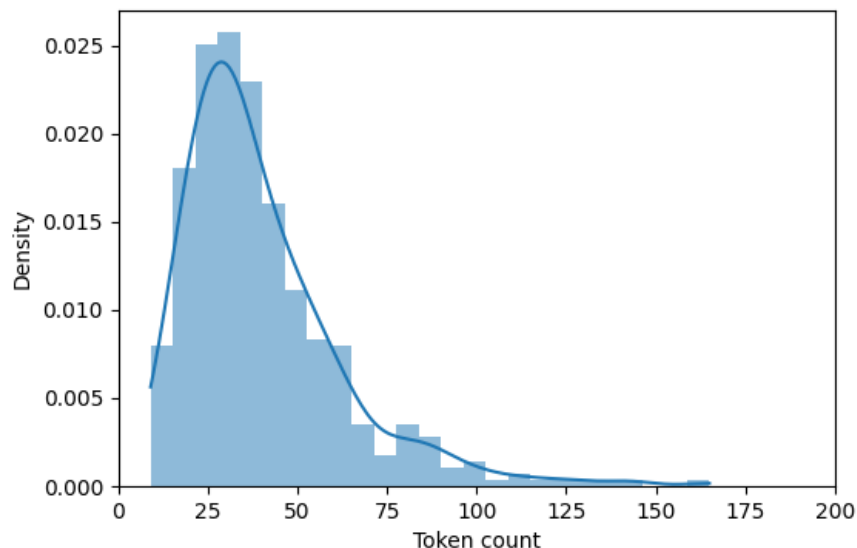

Figure 6. Token Distribution.

After knowing the length of the token in the model being tested, we then define the model structure shown in Figure 7. Based on the model in Figure 7, this model starts with an InputLayer layer that holds sequences of size 150, possibly representing text elements such as words. An Embedding layer follows, converting the integer index into a dense vector of size 100. This process then continues through a Conv1D layer with ReLU activation that uses 128 filters to produce a richer representation. Then, a MaxPooling1D layer reduces the spatial size, followed by a Bidirectional LSTM layer that allows the model to capture the context in both directions of the sequence data. The subsequent Dense layer with ReLU activation compresses the information into a 32-dimensional representation, while the Dropout layer helps prevent overfitting. Finally, the Dense layer with sigmoid activation produces a four-unit output suitable for binary or multi-label classification tasks. Overall, this architecture is designed to understand the context in sequence data by combining local feature extraction and understanding long-term dependencies, which is ideal for applications such as natural language processing or time series analysis.

Based on the model evaluation results using the CNN + Bi-LSTM algorithm it is presented in Table 5. Table 5 shows the performance of four different configurations in multi-label classification. Each configuration depicts the evaluation results of the CNN + Bi-LSTM model with data splits of 60:40, 70:30, 80:20, and 90:10 for training and testing. The evaluation results in the table include Accuracy, Precision, Recall, and F1-Score metrics. The configuration with 80:20 data sharing achieved the highest accuracy rate of 68.90%, with Precision, Recall, and F1-Score values reaching 66.20%, 71.40%, and 68.70% respectively. Meanwhile, the 70:30 and 90:10 configurations show almost similar performance, with Accuracy values ranging from 63.80% to 67.20%, and Precision, Recall, and F1-Score values that are in a range close to 61.00% to 64.50%, 66.70% to 70.00%, and 63.70% to 67.30%, respectively. The 60:40 configuration displays performance in the middle between the other configurations, with evaluation metric values of around 65.50% for Accuracy, 62.75% for precision, 68.25% for recall, and 65.00% for F1-Score. These results show how dividing training and testing data affects model performance in multi-label classification using CNN + Bi-LSTM.

Table 5. CNN+Bi-LSTM Evaluation Results

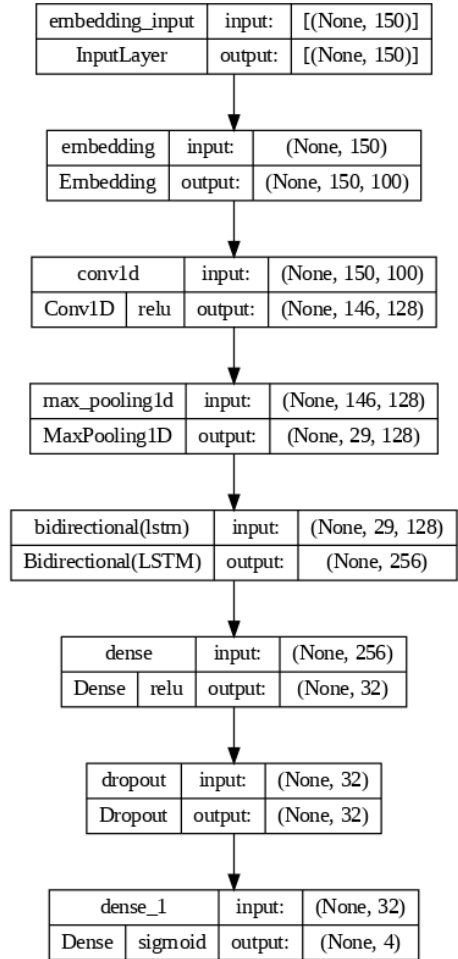| Configuration	Accuracy	Precision	Recall	F1-Score
60:40	65.50%	62.75%	68.25%	65.00%
70:30	63.80%	61.00%	66.70%	63.70%
80:20	68.90%	66.20%	71.40%	68.70%
90:10	67.20%	64.50%	70.00%	67.30%

Figure 7. CNN + Bi-LSTM Model Structure.

3.3. FastText Embedding

Representing words as vectors is expected to increase the accuracy of the CNN+Bi-LSTM model. Table 1 shows the test configuration for FastText word embedding, and Table 6 shows the test results for FastText word embedding.

Table 6. Accuracy Results of CNN + Bi-LSTM + FastText

FastText Parameter		Configuration			
Dimensions	Epoch	60:40	70:30	80:20	90:10
100	20	67.63%	66.99%	71.91%	70.54%
	50	66.85%	66.64%	63.29%	73.37%
	100	66.80%	66.10%	68.34%	66.81%
200	20	66.29%	69.06%	69.16%	68.41%
	50	68.56%	68.94%	70.72%	70.71%
	100	66.24%	68.47%	71.82%	72.31%
500	20	66.71%	70.95%	70.26%	72.84%
	50	66.57%	71.01%	73.46%	65.75%
	100	66.61%	69.83%	70.90%	72.13%

Table 6 shows the evaluation accuracy results of the CNN + Bi-LSTM + FastText model in four test configurations. The model evaluation results are based on the combination of FastText (Dimension) parameters with values 100, 200, and 300 and the Epoch variable with values 20, 50, and 100. Each parameter combination is evaluated on four different data split configurations: 60:40, 70: 30, 80:20, and 90:10.

For example, for a model with Dimension 100 and Epoch 20, the 60:40 configuration achieves an accuracy of 67.63%, while the 70:30 configuration has an accuracy of 66.99%. In Dimension 200 and Epoch 50, the 80:20 configuration shows a significant increase, with accuracy reaching 70.72%, while in the same configuration with Dimension 300 and Epoch 100, accuracy increases to 73.46%.

Meanwhile, the evaluation results with F1-Score are shown in Table 7. The evaluation results of the CNN + Bi-LSTM + FastText model are shown as F1-Score for each combination of parameters and configurations. For example, for a model with Dimension 100 and Epoch 20, the 80:20 configuration shows the highest F1-Score of 70.91%, while the 60:40 configuration has an F1-Score of 66.63%. In Dimension 300 and Epoch 50, the 70:30 configuration displays the highest F1-Score, namely 70.03%, while in the same configuration with Dimension 200 and Epoch 100, the F1-Score reaches 73.30%.

The results of the F1-Score evaluation provide an overview of model performance based on a combination of FastText parameters and data-sharing schemes. Changing values in each parameter combination and configuration can significantly impact the F1-Score, providing insight into how parameter configurations influence model evaluation results in the context of using CNN+Bi-LSTM augmented with FastText.

Table 7. CNN F1-score results + Bi-LSTM + FastText

FastText Parameter		Configuration			
Dimensions	Epoch	Dimensions	Epoch	Dimensions	Epoch
100	20	66.63%	65.99%	70.91%	69.54%
	50	65.87%	65.66%	62.31%	72.39%
	100	67.79%	67.09%	69.33%	67.80%
200	20	65.29%	68.06%	68.16%	67.41%
	50	67.58%	67.96%	69.74%	69.73%
	100	67.23%	69.46%	72.81%	73.30%
500	20	65.71%	69.95%	69.26%	71.84%
	50	65.59%	70.03%	72.48%	64.77%
	100	67.60%	70.82%	71.89%	73.12%

The use of FastText in this research appears to significantly impact the performance of the CNN+Bi-LSTM classification model in Al-Qur'an translation analysis. Table 5, which shows the evaluation results of the CNN+Bi-LSTM model without FastText, displays solid performance with F1-Score values ranging from 63.70% to 68.70% in various configurations. However, with the addition of FastText in Table 7, there is a consistent increase in the F1-Score value, with the highest value of 73.30% in dimension 200 and epoch 100. F1-Score is a combined measure of precision (Precision) and recall for most combinations of FastText parameters and configurations.

The results of this study show a significant improvement compared to two previous studies, namely [42] and [43], which used basic machine learning models to classify Al-Qur'an translations. Previously, research achieved 64.10% accuracy, 65% precision, and 62.68% recall. In the same context, the use of algorithms such as Support Vector Machine (SVM), Naïve Bayes (NB), and K-Nearest Neighbors (K-NN) in multi-label classification of Al-Qur'an translations, with an increasing amount of testing data, has been improved accuracy significantly. The results of testing using Support Vector Machine (SVM) show the most striking improvement with an accuracy of 70%.

The integration of FastText in the classification model has provided significant improvements in understanding and classifying Al-Qur'an translations, as evidenced by the increase in F1-Score values, which show a better balance between precision and recall in identifying labels such as Tauhid, Ibadah, Akhlaq, and History. However, the model has not

achieved 100% accuracy in classifying several Al-Qur'an translation labels. This may be due to the complexity of the language and variations in the context of the Quran, which are difficult to fully represent by machine language models. Varied sentence structures and different writing styles can also make language modeling difficult. Limitations in the training data set also affect the accuracy of predictions. In these complex contexts, achieving 100% accuracy is difficult due to the complexity of the language and the variations in interpretation that may occur.

4. CONCLUSION

This research shows a significant improvement in the multi-label classification of Indonesian Al-Qur'an translations. Using FastText in the CNN+Bi-LSTM classification model shows a consistent increase in F1-Score values, with the highest peak of 73.30%. The evaluation indicates that the incorporation of FastText has improved understanding of labels such as *Tauhid*, *Ibadah*, *Akhlak*, and *Sejarah*, as well as improved the classification of labels such as History and Tauhid. However, the quest for impeccable accuracy is impeded by the intricate linguistic features of the Qur'anic text and the diversity of its interpretations. Additionally, the model's performance is further constrained by the sentence structure and the scope of the training dataset. Although integrating FastText has led to notable improvements in dealing with complex textual content, achieving complete accuracy is challenging due to the complex nature of language and the wide range of possible interpretations that machine learning models have yet to fully capture.

Future research endeavors could focus on the enrichment of the training corpus with a more diverse set of interpretations, potentially enhancing the model's robustness and generalization capabilities. Moreover, exploring advanced ensemble methods that combine the strengths of multiple machine learning architectures may offer a pathway to further improvements in classification performance. Additionally, the application of transfer learning techniques, where models are pre-trained on extensive text corpora before fine-tuning on domain-specific datasets, might also contribute to a more nuanced understanding of the Qur'anic language and its translation challenges.

REFERENCES

[1]   A. Mokrani, "Islamic Hermeneutics of Nonviolence: Key Concepts and Methodological Steps," *Religions*, vol. 13, no. 4, p. 295, Mar. 2022, doi: 10.3390/rel13040295.

[2]   M. A. Ahmed, H. Baharin, and P. N. Ellyza Nohuddin, "K-means variations analysis for translation of English Tafseer Al-Quran text," *Int. J. Electr. Comput. Eng.*, vol. 13, no. 3, p. 3255, Jun. 2023, doi: 10.11591/ijece.v13i3.pp3255-3265.

[3]   M. Fauzan, H. Junaedi, and E. Setyati, "KLASIFIKASI AL – QUR'AN TERJEMAHAN BAHASA INDONESIA DENGAN MENGGUNAKAN ALGORITMA SUPPORT VECTOR MACHINE (SVM)," *KONVERGENSI*, vol. 18, no. 2, pp. 42–49, Dec. 2022, doi: 10.30996/konv.v18i1.6912.

[4]   M. H. Bashir *et al.*, "Arabic natural language processing for Qur'anic research: a systematic review," *Artif. Intell. Rev.*, vol. 56, no. 7, pp. 6801–6854, Jul. 2023, doi: 10.1007/s10462-022-10313-2.

[5]   E. Junianto and R. Rachman, "Implementation of Text Mining Model to Emotions Detection on Social Media Comments Using Particle Swarm Optimization and Naive Bayes Classifier," in *2019 7th International Conference on Cyber and IT Service Management (CITSM)*, Nov. 2019, pp. 1–6. doi: 10.1109/CITSM47753.2019.8965382.

[6]   Y. Ying, T. N. Mursitama, Shidarta, and Lohansen, "Effectiveness of the News Text Classification Test Using the Naïve Bayes' Classification Text Mining Method," *J. Phys. Conf. Ser.*, vol. 1764, no. 1, p. 012105, Feb. 2021, doi: 10.1088/1742-6596/1764/1/012105.

[7]   B. M. P. Waseso and N. A. Setiyanto, "Web Phishing Classification using Combined Machine Learning Methods," *J. Comput. Theor. Appl.*, vol. 1, no. 1, pp. 11–18, Aug. 2023, doi: 10.33633/jcta.v1i1.8898.

[8]   B. Yang *et al.*, "Automatic Text Classification for Label Imputation of Medical Diagnosis

Notes Based on Random Forest," 2018, pp. 87–97. doi: 10.1007/978-3-030-01078-2_8.

[9]   Y. Fang and P. Zhang, "Recognition of Spam Messages Based on Text Mining," *J. Phys. Conf. Ser.*, vol. 1624, no. 5, p. 052024, Oct. 2020, doi: 10.1088/1742-6596/1624/5/052024.

[10]   F. Mustofa, A. N. Safriandono, and A. R. Muslikh, "Dataset and Feature Analysis for Diabetes Mellitus Classification using Random Forest," *J. Comput. Theor. Appl.*, vol. 1, no. 1, pp. 41–48, Sep. 2023, doi: 10.33633/jcta.v1i1.9190.

[11]   M. I. Akazue, I. A. Debekeme, A. E. Edje, C. Asuai, and U. J. Osame, "UNMASKING FRAUDSTERS: Ensemble Features Selection to Enhance Random Forest Fraud Detection," *J. Comput. Theor. Appl.*, vol. 1, no. 2, pp. 201–211, Dec. 2023, doi: 10.33633/jcta.v1i2.9462.

[12]   Nur Ghaniaviyanto Ramadhan, "Indonesian Online News Topics Classification using Word2Vec and K-Nearest Neighbor," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 5, no. 6, pp. 1083–1089, 2021, doi: 10.29207/resti.v5i6.3547.

[13]   C. Wei, P. Zhou, D. Li, Y. Zhong, and Y. Han, "Application of SA-KNN combined with SVM based on different kernel functions in pattern knowledge mining," in *International Conference on Computer Graphics, Artificial Intelligence, and Data Processing (ICCAID 2021)*, Mar. 2022, p. 23. doi: 10.1117/12.2631116.

[14]   J. Hermanto, "Klasifikasi Teks Humor Bahasa Indonesia Memanfaatkan SVM," *J. Inf. Syst. Hosp. Technol.*, vol. 3, no. 01, pp. 39–48, 2021, doi: 10.37823/insight.v3i01.118.

[15]   K. I. Gunawan and J. Santoso, "Multilabel Text Classification Menggunakan SVM dan Doc2Vec Classification Pada Dokumen Berita Bahasa Indonesia," *Journal of Information System,Graphics, Hospitality and Technology*, vol. 3, no. 01. pp. 29–38, 2021. doi: 10.37823/insight.v3i01.126.

[16]   A. Hanafi, A. Adiwijaya, and W. Astuti, "Klasifikasi Multi Label pada Hadis Bukhari Terjemahan Bahasa Indonesia Menggunakan Mutual Information dan k-Nearest Neighbor," *J. Sisfokom (Sistem Inf. dan Komputer)*, vol. 9, no. 3, pp. 357–364, 2020, doi: 10.32736/sisfokom.v9i3.980.

[17]   R. Hayami, S. Mohnica, and Soni, "Klasifikasi multilabel komentar toxic pada sosial media twitter menggunakan convolutional neural network(CNN)," *J. CoSciTech (Computer Sci. Inf. Technol.*, vol. 4, no. 1, pp. 1–6, 2023, doi: 10.37859/coscitech.v4i1.4365.

[18]   I. F. Rozi, V. N. Wijayaningrum, and N. Khozin, "Klasifikasi Teks Laporan Masyarakat Pada Situs Lapor! Menggunakan Recurrent Neural Network," *Sistemasi*, vol. 9, no. 3, p. 633, 2020, doi: 10.32520/stmsi.v9i3.977.

[19]   A. Omar, T. M. Mahmoud, T. Abd-El-Hafeez, and A. Mahfouz, "Multi-label Arabic text classification in Online Social Networks," *Inf. Syst.*, vol. 100, p. 101785, 2021, doi: 10.1016/j.is.2021.101785.

[20]   W. Kurnia Sari, D. Palupi Rini, R. Firsandaya Malik, and I. B. Saladin Azhar, "Klasifikasi Teks Multilabel pada Artikel Berita Menggunakan Long Short-Term Memory dengan Word2Vec," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 1, no. 3, pp. 276–285, 2017.

[21]   S. Sayyida, "Ayat-Ayat Tauhid Terhadap Budaya Pemeliharaan Keris Di Jawa (Studi Kasus Buku Mt Arifin)," *J. Qur'an Hadith Stud.*, vol. 6, no. 1, pp. 24–52, 2019, doi: 10.15408/quhas.v6i1.13403.

[22]   A. Kallang, "Konteks Ibadah Menurut Al-Quran," *Al-Din J. Dakwah dan Sos. Keagamaan*, vol. 4, no. 2, pp. 1–13, 2018, doi: 10.35673/ajdsk.v4i2.630.

[23]   M. Murharyana, I. I. Al Ayyubi, and R. Rohmatulloh, "Pendidikan Akhlak Anak Kepada Orang Tua Dalam Perspektif Al-Quran," *Piwulang J. Pendidik. Agama Islam*, vol. 5, no. 2, pp. 175–191, 2023, doi: http://dx.doi.org/10.32478/piwulang.v5i2.1515.

[24]   J. Mirdad and S. Rahmat, "Sejarah Dalam Perspektif Islam," *El -Hekam*, vol. 6, no. 1, p. 9, 2021, doi: 10.31958/jeh.v6i1.3303.

[25]   S. Bessou and R. Aberkane, "Subjective Sentiment Analysis for Arabic Newswire Comments," *J. Digit. Inf. Manag.*, vol. 17, no. 5, p. 289, 2019, doi: 10.6025/jdim/2019/17/5/289-295.

[26] A. A. Firdaus, A. Yudhana, and I. Riadi, "Public Opinion Analysis of Presidential Candidate Using Naïve Bayes Method," *Kinet. Game Technol. Inf. Syst. Comput. Network, Comput. Electron. Control*, vol. 4, no. 2, 2023, doi: 10.22219/kinetik.v8i2.1686.

[27] E. Y. Sari, A. D. Wierfi, and A. Setyanto, "Sentiment Analysis of Customer Satisfaction on Transportation Network Company Using Naive Bayes Classifier," *2019 Int. Conf. Comput. Eng. Network, Intell. Multimedia, CENIM 2019 - Proceeding*, vol. 2019-Novem, 2019, doi: 10.1109/CENIM48368.2019.8973262.

[28] A. Alwehaibi, M. Bikdash, M. Albogmi, and K. Roy, "A study of the performance of embedding methods for Arabic short-text sentiment analysis using deep learning approaches," *J. King Saud Univ. - Comput. Inf. Sci.*, no. xxxx, 2021, doi: 10.1016/j.jksuci.2021.07.011.

[29] Y. Taher, A. Moussaoui, and F. Moussaoui, "Automatic Fake News Detection based on Deep Learning, FastText and News Title," *Int. J. Adv. Comput. Sci. Appl.*, vol. 13, no. 1, pp. 146–158, 2022, doi: 10.14569/IJACSA.2022.0130118.

[30] H. A. Almuzaini and A. M. Azmi, "Impact of Stemming and Word Embedding on Deep Learning-Based Arabic Text Categorization," *IEEE Access*, vol. 8, pp. 127913–127928, 2020, doi: 10.1109/ACCESS.2020.3009217.

[31] D. I. Af'idah, D. Dairoh, S. F. Handayani, and R. W. Pratiwi, "Pengaruh Parameter Word2Vec terhadap Performa Deep Learning pada Klasifikasi Sentimen," *J. Inform. J. Pengemb. IT*, vol. 6, no. 3, pp. 156–161, 2021, doi: 10.30591/jpit.v6i3.3016.

[32] E. Lim, E. I. Setiawan, and J. Santoso, "Stance Classification Post Kesehatan di Media Sosial Dengan FastText Embedding dan Deep Learning," *J. Intell. Syst. Comput.*, vol. 1, no. 2, pp. 65–73, 2019, doi: 10.52985/insyst.v1i2.86.

[33] A. Saifudin, "Metode Data Mining Untuk Seleksi Calon Mahasiswa," vol. 10, no. 1, pp. 25–36, 2018, doi: https://dx.doi.org/10.24853/jurtek.10.1.25-36.

[34] Yudi Widhiyasana, Transmissia Semiawan, Ilham Gibran Achmad Mudzakir, and Muhammad Randi Noor, "Penerapan Convolutional Long Short-Term Memory untuk Klasifikasi Teks Berita Bahasa Indonesia," *J. Nas. Tek. Elektro dan Teknol. Inf.*, vol. 10, no. 4, pp. 354–361, 2021, doi: 10.22146/jnteti.v10i4.2438.

[35] S. Ali, A. Hashmi, A. Hamza, U. Hayat, and H. Younis, "Dynamic and Static Handwriting Assessment in Parkinson ' s Disease : A Synergistic Approach with C-Bi-GRU and VGG19," *J. Comput. Theor. Appl.*, vol. 1, no. 2, pp. 151–162, 2023, doi: 10.33633/jcta.v1i2.9469.

[36] D. T. Hermanto, A. Setyanto, and E. T. Luthfi, "Algoritma LSTM-CNN untuk Binary Klasifikasi dengan Word2vec pada Media Online," *Creat. Inf. Technol. J.*, vol. 8, no. 1, p. 64, 2021, doi: 10.24076/citec.2021v8i1.264.

[37] H. Deng, D. Ergu, F. Liu, Y. Cai, and B. Ma, "Text sentiment analysis of fusion model based on attention mechanism," *Procedia Comput. Sci.*, vol. 199, pp. 741–748, 2021, doi: 10.1016/j.procs.2022.01.092.

[38] A. M. Ertugrul and P. Karagoz, "Movie Genre Classification from Plot Summaries Using Bidirectional LSTM," *Proc. - 12th IEEE Int. Conf. Semant. Comput. ICSC 2018*, vol. 2018-Janua, pp. 248–251, 2018, doi: 10.1109/ICSC.2018.00043.

[39] N. N. Wijaya, D. R. Ignatius, M. Setiadi, and A. R. Muslikh, "Music-Genre Classification using Bidirectional Long Short- Term Memory and Mel-Frequency Cepstral Coefficients," *J. Comput. Theor. Appl.*, vol. 2, no. 1, pp. 13–26, 2024, doi: 10.33633/jcta.v2i1.9655.

[40] A. R. Isnain, A. Sihabuddin, and Y. Suyanto, "Bidirectional Long Short Term Memory Method and Word2vec Extraction Approach for Hate Speech Detection," *IJCCS (Indonesian J. Comput. Cybern. Syst.*, vol. 14, no. 2, p. 169, 2020, doi: 10.22146/ijccs.51743.

[41] A. Alfando and R. Hayami, "Klasifikasi Teks Berita Berbahasa Indonesia Menggunakan Machine Learning Dan Deep Learning: Studi Literatur," *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 7, no. 1. pp. 681–686, 2023. doi: 10.36040/jati.v7i1.6486.

[42] T. H. Putrisanni, A. Adiwijaya, and S. Al Faraby, "Klasifikasi Ayat Al-Quran Terjemahan

Bahasa Inggris Menggunakan K-Nearest Neighbor (Knn) Dan Information Gain," *KOMIK (Konferensi Nas. Teknol. Inf. dan Komputer)*, vol. 3, no. 1, pp. 362–369, 2019, doi: 10.30865/komik.v3i1.1614.

[43] A. Abdullahi, N. A. Samsudin, M. H. A. Rahim, S. K. A. Khalid, and R. Efendi, "Multi-label classification approach for Quranic verses labeling," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 24, no. 1, pp. 484–490, 2021, doi: 10.11591/ijeecs.v24.i1.pp484-490.

● **20% Overall Similarity**

Top sources found in the following databases:

- 15% Internet database
- Crossref database

- 11% Publications database
- Crossref Posted Content database

TOP SOURCES

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	**publikasi.dinus.ac.id** Internet	**4%**
2	**Hongjie Deng, Daji Ergu, Fangyao Liu, Ying Cai, Bo Ma. "Text sentiment ...** Crossref	**2%**
3	**ia601607.us.archive.org** Internet	**1%**
4	**jutif.if.unsoed.ac.id** Internet	**1%**
5	**Oddy Virgantara Putra, Fajar Deny Elfianita, Triana Harmini, Asif Trisna...** Crossref	**<1%**
6	**ebin.pub** Internet	**<1%**
7	**mdpi.com** Internet	**<1%**
8	**repository.uin-malang.ac.id** Internet	**<1%**
9	**ijisae.org** Internet	**<1%**

22 ejurnal.stie-trianandra.ac.id
Internet
<1%

23 Brian Rizqi Paradisiaca Darnoto, Daniel Siahaan, Diana Purwitasari. "De...
Crossref
<1%

24 escholarship.org
Internet
<1%

25 sciencegate.app
Internet
<1%

26 Manoj. V, Devi. T, M. Kalaiyarasi. "Classification of Newspaper Article ...
Crossref
<1%

27 ejurnal.stmik-budidarma.ac.id
Internet
<1%

28 Akbar Ridwan, Hilal H. Nuha, Ramanti Dharayani. "Sentiment Analysis ...
Crossref
<1%

29 Farhan Dzaky Arvianto, Moch Arif Bijaksana, Arief Fatchul Huda. "Peop...
Crossref
<1%

30 Gaddisa Olani Ganfure. "Comparative analysis of deep learning based ...
Crossref
<1%

31 Qomarudin Sifak, Erwin Budi Setiawan. "Hate Speech Detection using ...
Crossref
<1%

32 ijece.iaescore.com
Internet
<1%

33 www2.mdpi.com
Internet
<1%

46   Xiuwei Yan, Sijia Liu, Songlei Wang, Jiarui Cui et al. "Predictive Analysi...   <1%
      Crossref