Building of Informatics, Technology and Science (BITS)
Volume 4, No 3, Desember 2022 Page: 1548−1557
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v4i3.2718

Hate Speech Detection on Twitter through Natural Language Processing using LSTM Model

Cepthari Ningtyas Arbaatun*, Dade Nurjanah, Hani Nurrahmi

Informatic, School of Computing, Telkom University, Bandung, Indonesia
Email: [1,*]ceptarityasa@student.telkomuniversity.ac.id, [2]dadenurjanah@telkomuniversity.ac.id,
[3]haninurrahmii@telkomuniversity.ac.id
Correspondence Author Email: ceptarityasa@student.telkomuniversity.ac.id
Submitted: **20/12/2022**; Accepted: **28/12/2022**; Published**: 30/12/2022**

Abstract− Currently, social media is a place to express opinions. This opinion can be positive or negative. However, lately, the opinion that often appears is a negative opinion, such as hate speech. Hate speech is often found on social media, such as malicious comments intended to insult individuals or groups. Based on WeAreSocial data in 2021, one of the most used social media platforms in Indonesia is Twitter, with 63.6% of users. According to the Indonesia National Police, hate speech cases were more dominant during the period from April 2020 to July 2021. Combating cybercriminals is also difficult, therefore infrastructure and personnel are required. Therefore, efforts are needed to identify hate speech on the Twitter platform in Indonesian so that law enforcement can detect the spread of hate speech. Deep learning is one method for detecting hate speech. In this research, we use a deep learning model of Long Short-Term Memory (LSTM) with word embedding. FastText and Global Vector (GloVe) is the word embeddings that we use as input for word representation and classification. FastText embeddings make use of subword information to create word embeddings and GloVe embeddings using an unsupervised learning method trained on a corpus to generate distributional feature vectors. From the evaluation results on the experimental model, LSTM-FastText using random oversampling has an advantage with an F1-score of 89.91% compared to LSTM-GloVe to obtain an F1-score of 82.14%.

Keywords: Hatespeech; Twitter; Fasttext; Glove; Lstm

1. INTRODUCTION

In recent years, there has been a rapid increase in the use of social media such as Facebook, Twitter, Instagram, and others. Based on WeAreSocial data in 2021, one of Indonesia's most used social media platforms is Twitter, with a value of 63.6% of users [1]. On Twitter, users can freely tweet, upload photos, and share information on their accounts, including creating tweets containing hate speech.

According to Paula and Sérgio [2], hate speech is a language that attacks or demeans that incites violence or hatred against groups based on specific characteristics such as physical appearance, religion, descent, national or ethnic origin, sexual orientation, gender identity or other, and it can occur in a variety of linguistic styles, including subtly or with humor. One of the hate speeches that has received much attention is those directed at public officials, religious leaders, and public figures [3]. The Indonesian National Police [4] reported that hate speech cases dominated pornographic content reports from April 2020 to July 2021. Namely, there were around 473 cases, including provocative cases, hate content, and hate speech. The Indonesia National Police stated that combating cybercriminals is not simple, so infrastructure and personnel are required [5]. The impact of the problem of hate speech on social media can become considerable and widespread, with harmful implications for people, communities, and society if it is not addressed appropriately. This necessitates the development of an automatic detection tool for hate speech in the Indonesian language so that law enforcement can detect the spread of hate speech. Therefore, to solve the issue of hate speech, one of the solutions is to detect hate speech on social media using deep learning, which is part of machine learning and functions to train computers about basic human instincts.

Research [6] experimented with toxic comment classification by taking two datasets, the Google Jigsaw and Twitter datasets [7]. The purpose of the research [6] was to compare deep learning and propose an ensemble for individual classifiers in the F1-score. Deep learning carried out includes using LSTM-FastText and LSTM-GloVe. From the comparison results, LSTM-GloVe obtained an F1-score of 78.1 and was ahead of LSTM-FastText by 77.8%. In comparison, the ensemble is far superior, with an F1-score of 79.3%. However, the error in the analysis of this study is that the ensemble results identified subtasks that were difficult to classify as toxic comments because there needed to be more consistent label quality. In addition, unsolved challenges occur due to needing more training data with very special or rare vocabulary.

In research [8], [9] participated in the Hate Speech Detection on Social Networks organized by VLSP Shared 2019 with the aim of detecting Vietnamese social media text according to predetermined labels. The label consists of a pre-label dataset and an unlabeled dataset for comments or social media posts. The downside of such datasets is that the language is a low resource for natural language processing. Research [8] pre-processed and built machine learning models to classify comments or posts. Using two-word embeddings as a comparison, we get the best word embeddings, which are FastText and GloVe. The models used include SVM, Logistic Regression, GRU, and Bidirectional-LSTM. When experimenting with word embedding GloVe with the baomoi.vn.model dataset.txt, accuracy, precision, memory, and F1-score levels of 93.26%, 90.74%, 50.30%, and 53.62%, respectively, were obtained in the training dataset. Similar to word embedding, FastText gets accuracy, precision, memory, and F1-scores of 95.67%, 85.61%, 67.36%, and 73.84%, respectively, on the training dataset. And combining Bi-LSTM with

Building of Informatics, Technology and Science (BITS)
Volume 4, No 3, Desember 2022 Page: 1548−1557
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v4i3.2718

FastText will bring better results. So, for the dataset of VLSP Shared 2019, it gets an F1-Score of 71.43%. Deep learning methods based on the Bi-GRU-LSTM-CNN classifier with word embedding FastText as pre-training are used in research [9]. The study obtained an F1-score of 70.576%.

Research by [10] conducted a hate speech detection experiment on Twitter with a dataset of 16 thousand tweets that other studies have annotated. For word embedding, use random embedding and GloVe. As for optimization, it uses 'Adam' for CNN and LSTM and 'RMS-Prop' for FastText. Word embedding learned from deep neural network (DNN) models, combined with the Gradient Boosted Decision tree(GBDT), yielded the best accuracy value for LSTM-Random Embedding-GBDT with an F1-score of 93% compared to FastText-GloVe F1-score of 82.9% and LSTM-GloVe of 80.8%. In addition, similar words obtained using DNN learning embeddings clearly show "resentment" towards the target words, which generally are not seen in similar words obtained using GloVe.

This research uses hate speech with elements of blasphemy, provocation, and incitement. The blasphemy element was added because the Holywings company posted posters on social media with free liquor for the end of the day named Muhammad and Maria [11]. In addition to Holywings, Netflix aired the film The Umbrella Academy 3, in which Lafaz Allah appears to be written on the floor, and one of the characters is standing on the same floor [12]. Companies originating from China sell bikinis with the design of Quranic verses [13]. Later, the insult of the Prophet Muhammad was also carried out by the Bharatiya Janata Party (BJP) in India during a television debate on the Gyanvapi Mosque. The calls for a boycott of Holywings, Netflix, China, and India. The element of provocation and incitement was taken from the foundation for distributing donations for Aksi Cepat Tanggap (ACT) for allegedly misappropriating people's funds [14] and followed by the Ministry of Social Affairs, which revoked ACT because it was judged that there was a violation of the Minister of Social Affairs Regulation [15]. It became trending on Twitter with #KamiPercayaACT and #JanganPercayaACT.

Based on the existing problems, this research will discuss the performance value in the classification of hate speech identification on Twitter using LSTM-FastText and LSTM-GloVe. As well as the value of the influence of unbalanced and balanced data on the LSTM-FastText and LSTM-GloVe methods. The purpose of the problem is to find out the results of comparing the accuracy, precision, recall, and F1-score values of the LSTM model with word embedding GloVe and LSTM with word embedding FastText in classifying hate speech text. In addition, to analyze the value of the influence of unbalanced and balanced data on the LSTM-FastText and LSTM-GloVe methods.

2. RESEARCH METHODOLOGY

2.1 Research Stages

In this research, the system built for classifying hate speech on Twitter is carried out in 6 steps. According to Figure 1, the system's flow begins by collecting Twitter data and saving it in Excel in .csv format. Furthermore, the data is labelled manually. The data is then processed at the preprocessing stage so that the data is structured and uniform. Next is word embedding, which involves representing the words in a tweet as a numeric vector. Then the data is classified using the LSTM approach with word embedding and evaluated to get accuracy and F1-score results. Figure 1. Overviews the system architecture design used to conduct this research.

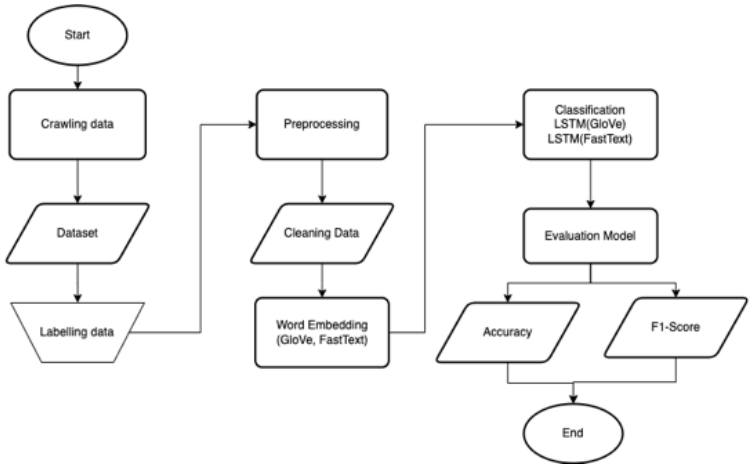

Figure 1. System Architecture Design

2.2 Dataset

The dataset for this research is crawling data from Twitter. Retrieval of tweets on Twitter requires an API key to access Twitter data. At the time of Twitter data collection, researchers used nine hashtags and six keywords containing blasphemy, provocation, incitement, and explicit words to detect hate speech. As one of the characteristics of hate speech, explicit words are also set to be topics. According to Table 1. The researcher was able to collect 3253 tweets

Building of Informatics, Technology and Science (BITS)
Volume 4, No 3, Desember 2022 Page: 1548−1557
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v4i3.2718

about blasphemy with 86 tweets, provocation and incitement with 512 tweets, and explicit terms with a total of 2655 tweets via crawling data.

Table 1. Crawling Hashtag and Keyword List

Topic	Hashtags	Total
Penistaan Agama	#boikotholywings, #tutupholywings, #boikotumbrella, #boikotindia, #boikotchina	86
Provokasi dan Hasutan	#kamipercayaact, #janganpercayaact, #aksicepattilep, #aksicepattanggap, Kemensos	512
Kata-kata Eksplisit	anjing, goblok, tolol, bajingan	2655

2.3 Data Annotations

Every tweet in the dataset is labelled whether it contains hate speech or not. In the labelling process, there are two data labels. Tweets that contain hate speech will be labelled "HS", and those that do not contain non-hate speech are labelled as "Non_HS". The data labelling process was done manually by five volunteers with specifications for having used Twitter and in terms of ages 21 to 28 years to determine tweets categorized as HS or Non_HS from the results of data crawling. Because volunteers carry out the labelling process, several tweets are different, and voting is carried out to determine whether the tweets are labelled HS or Non_HS. Of the 3253 tweets, 2344 tweets contain as HS and 909 tweets as Non_HS. Because the number of HS label tweets is different from the number of Non_HS labels, the results of the dataset become imbalanced data. The labeling of HS and Non_HS tweets is represented in Table 2. below.

Table 2. Data Labelling

Label	Tweet
HS	b'@CNNIndonesia Gak nyangka sekelas Mentri Kemensos korupsi bansos dasar biadab gak beradab.....terlaknat...'
Non_HS	kami akan sentiasa bangkit utk pertahankan Nabi kami, tuhan kami(Allah) dan juga agama kami Islam #islambangkit #TUTUPHOLYWINGS #BoikotNetflix #BoikotIndia #boikotchina #boikotacademyumbrella #shameonisrael

2.4 Preprocessing

At this stage of preprocessing. First, after preprocessing, the data findings are saved in the.csv format in Excel. Then, the data is evaluated to determine whether it is qualified or not. If the data is of excellent quality, the following procedure can be continued. If the data is not of good quality, then the data needs to be cleaned manually through Excel. The flow of the preprocessing stage can be seen in Figure 2. below.

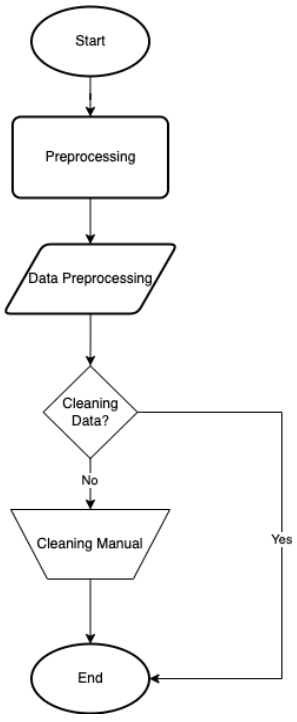

Figure 2. Preprocessing Diagram Flow

Building of Informatics, Technology and Science (BITS)
Volume 4, No 3, Desember 2022 Page: 1548−1557
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v4i3.2718

2.5 FastText

Word embedding introduced by Bojanowski et al. [16] is FastText. FastText is a library for efficient word representation and sentence classification. FastText embedding uses subword information to create word embeddings. N-gram character representations are studied, and words are represented as sums of n-gram vectors. This adds subword information to the word2vec type model. This facilitates the embeddings and understanding of suffixes and prefixes. Once a word is represented using n-gram characters, the skip-gram model is trained to learn its embeddings.

2.6 GloVe

In research [17] offers a well-known additional word insertion model called GloVe (Global Vector for Word Representation). GloVe discovered embeddings using unsupervised learning methods trained on the corpus to generate vectors of distribution features. During the learning process, a statistical-based matrix is built to represent the appearance of words in the corpus. This matrix displays vector terms. The learning process takes time and space for matrix development, which is a very expensive procedure. The gloVe is a count-based model, while Word2Vec is a prediction-based approach. The gloVe is studied using Wikipedia, web data, and Twitter, with different vector dimensions accessible for each model [17].

2.7 LSTM

Another type of neural network is a recurrent neural network (RNN), which performs the same task for each element in the sequence, with the output depending on previous calculations. In time-dependent or sequential tasks, RNN outperforms CNN because it maps the entire history of previous inputs to each output. This recurring connection allows the node to have a previous input "memory" that affects the network output.

Long-Short Term Memory (LSTM) is a specific type of RNN capable of analyzing and studying long-term dependencies. LSTM is explicitly considered to address long-term dependency issues. They can easily remember information for as long as necessary, which is best when working on sentence order [18].

Due to its large memory capacity, LSTM is often preferred for text classification and predictive modelling tasks. Such networks deliberately choose what information to pass on to additional neurons and what information can be forgotten or abandoned. This network uses gated and backpropagation mechanisms. The LSTM network consists of an input gate(i_t),), an output gate(o_t), and a forget gate (f_t), which are presented in equations (1)–(3).

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i \quad (1)$$

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f \quad (2)$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o \quad (3)$$

Here, x_t is the input text, h is the input state, where h_t is the current state and h_{t-1} is the previous state. The weights and biases for each gate are W and b respectively. The activation function used here, σ indicated by the symbol, which in the case of the suggested model is Relu[19].

2.8 Evaluation

This study used model evaluation to determine the predicted results as the final stage carried out after classification. The results of the model evaluation use four specific precision metrics: accuracy, recall, precision, and F1-score. Accuracy is calculating the accuracy of a learning model by calculating the number of occurrences of a sample that corresponds to a predetermined set of values.

$$Accuracy = \frac{True\ Negative + True\ Positive}{True\ Negative + False\ Positive + True\ Positive + False\ Negative} \quad (4)$$

The Recall is the ratio of correct positive predictions compared to overall correct positive data.

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \quad (5)$$

The precision encapsulates the precision-recall curve as the average precision achieved at each threshold.

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \quad (6)$$

The F1-score can be understood as a weighted average of precision and memory, where the best F1-score is 1, and the worst is 0. Equivalent contributions are made by precision (E. 6) and recall (E. 7) to the F1-score.

$$F1\ score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (7)$$

Building of Informatics, Technology and Science (BITS)
Volume 4, No 3, Desember 2022 Page: 1548−1557
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v4i3.2718

3. RESULT AND DISCUSSION

3.1 Preprocessing Results

a. Lowercasing and removing special characters, punctuation, URLs

At the lowercasing stage, that is to change the text to lowercase. It is very important to lowercase text because a computer can evaluate the same word twice if it is written in uppercase or other cases. When text is vectorized for feature extraction, the words 'boikot' and 'Boikot', for example, can be evaluated differently and allocated different vectors. After successfully converting the text to lowercase, it removes special characters, punctuation marks, and URLs. The raw data contains many examples of punctuation marks or special characters (@, $, *,(),[],!,:,',#,\,/) that are not very important and are not understood by the computer. As a result, its presence in the data contributes to noise and must be eliminated. This is done by removing all punctuation marks and special characters using regular expressions. Regular expressions are also used to remove URLs from content that is not usually important. The results of lowercasing are shown in Table 3.

Table 3. Lowercasing and removing special character, punctuation, URLs

Before Lowercasing	After Lowercasing
b'@CNNIndonesia Gak nyangka sekelas Mentri Kemensos korupsi bansos dasar biadab gak beradab.....terlaknat...'	gak nyangka sekelas mentri kemensos korupsi bansos dasar biadab gak beradab terlaknat
kami akan sentiasa bangkit utk pertahankan Nabi kami, tuhan kami(Allah) dan juga agama kami Islam #islambangkit #TUTUPHOLYWINGS #BoikotNetflix #BoikotIndia #boikotchina #boikotacademyumbrella #shameonisrael	kami akan sentiasa bangkit utk peahankan nabi kami tuhan kamiallah dan juga agama kami islam islambangkit tutupholywings boikotnetflix boikotindia boikotchina boikotacademyumbrella shameonisrael

b. Tokenization

In the tokenization process, most break or divide sentences into sets of tokens or words. Depending on the criteria that apply, sentences are usually divided by a space between the two words or even punctuation. In Table 4. the results before and after tokenization are shown.

Table 4. Tweet Tokenization

Before Tokenization	After Tokenization
gak nyangka sekelas mentri kemensos korupsi bansos dasar biadab gak beradab terlaknat	['gak', 'nyangka', 'sekelas', 'mentri', 'kemensos', 'korupsi', 'bansos', 'dasar', 'biadab', 'gak', 'beradab', 'terlaknat']
kami akan sentiasa bangkit utk peahankan nabi kami tuhan kamiallah dan juga agama kami islam islambangkit tutupholywings boikotnetflix boikotindia boikotchina boikotacademyumbrella shameonisrael	['kami', 'akan', 'sentiasa', 'bangkit', 'utk', 'peahankan', 'nabi', 'kami', 'tuhan', 'kamiallah', 'dan', 'juga', 'agama', 'kami', 'islam', 'islambangkit', 'tutupholywings', 'boikotnetflix', 'boikotindia', 'boikotchina', 'boikotacademyumbrella', 'shameonisrael']

c. Slangword

In Indonesian, people often write non-standard words (slang words) in place of standard words, such as "cepet" for "cepat". If this happens, the computer will read "gak" and "tidak" as separate words, even though they mean the same thing. All slang words need to be replaced with more formal versions to fix this. These results are shown in Table 5.

Table 5. Change Slangword

Before Slangword	After Slangword
['gak', 'nyangka', 'sekelas', 'mentri', 'kemensos', 'korupsi', 'bansos', 'dasar', 'biadab', 'gak', 'beradab', 'terlaknat']	['tidak', 'menyangka', 'sekelas', 'menteri', 'kemensos', 'korupsi', 'bansos', 'dasar', 'biadab', 'tidak', 'beradab', 'terlaknat']
['kami', 'akan', 'sentiasa', 'bangkit', 'utk', 'peahankan', 'nabi', 'kami', 'tuhan', 'kamiallah', 'dan', 'juga', 'agama', 'kami', 'islam', 'islambangkit', 'tutupholywings', 'boikotnetflix', 'boikotindia', 'boikotchina', 'boikotacademyumbrella', 'shameonisrael']	['kami', 'akan', 'sentiasa', 'bangkit', 'untuk', 'peahankan', 'nabi', 'kami', 'tuhan', 'kamiallah', 'dan', 'juga', 'agama', 'kami', 'islam', 'islambangkit', 'tutupholywings', 'boikotnetflix', 'boikotindia', 'boikotchina', 'boikotacademyumbrella', 'shameonisrael']

d. Stemming

Stemming is the process of shortening a word and returning it to its root form. In some circumstances, the approach shortens the word to the point where the semantics are stored but the meaning is lost. Table 6. shows the results of stemming and one example of stemming is "menyangka" to "sangka".

Building of Informatics, Technology and Science (BITS)
Volume 4, No 3, Desember 2022 Page: 1548−1557
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v4i3.2718

Table 6. Stemming Word

Before Stemming	After Stemming
['tidak', 'menyangka', 'sekelas', 'menteri', 'kemensos', 'korupsi', 'bansos', 'dasar', 'biadab', 'tidak', 'beradab', 'terlaknat']	['tidak', 'sangka', 'kelas', 'menteri', 'kemensos', 'korupsi', 'bansos', 'dasar', 'biadab', 'tidak', 'adab', 'laknat']
['kami', 'akan', 'sentiasa', 'bangkit', 'untuk', 'peahankan', 'nabi', 'kami', 'tuhan', 'kamiallah', 'dan', 'juga', 'agama', 'kami', 'islam', 'islambangkit', 'tutupholywings', 'boikotnetflix', 'boikotindia', 'boikotchina', 'boikotacademyumbrella', 'shameonisrael']	['kami', 'akan', 'sentiasa', 'bangkit', 'untuk', 'peahankan', 'nabi', 'kami', 'tuhan', 'kamiallah', 'dan', 'juga', 'agama', 'kami', 'islam', 'islambangkit', 'tutupholywings', 'boikotnetflix', 'boikotindia', 'boikotchina', 'boikotacademyumbrella', 'shameonisrael']

e. Stopword Removal

Stopword removes low-information words from a text to focus on important words. NLP extracts keywords related to a specific topic depending on the use case. Words such as 'tidak,' 'yang,' 'tadi,' ' kami,' 'akan', and others are irrelevant for text classification and are frequently ignored. These words are referred to as stopwords and must be identified as soon as possible. The results for stopwords are shown in Table 7.

Table 7. Stopword Removal

Before	After
['tidak', 'sangka', 'kelas', 'menteri', 'kemensos', 'korupsi', 'bansos', 'dasar', 'biadab', 'tidak', 'adab', 'laknat']	['sangka', 'kelas', 'menteri', 'kemensos', 'korupsi', 'bansos', 'dasar', 'biadab', 'adab', 'laknat']
['kami', 'akan', 'sentiasa', 'bangkit', 'untuk', 'peahankan', 'nabi', 'kami', 'tuhan', 'kamiallah', 'dan', 'juga', 'agama', 'kami', 'islam', 'islambangkit', 'tutupholywings', 'boikotnetflix', 'boikotindia', 'boikotchina', 'boikotacademyumbrella', 'shameonisrael']	['sentiasa', 'bangkit', 'peahankan', 'nabi', 'tuhan', 'kamiallah', 'agama', 'islam', 'islambangkit', 'tutupholywings', 'boikotnetflix', 'boikotindia', 'boikotchina', 'boikotacademyumbrella', 'shameonisrael']

After the stopword is removed, it is returned to a regular sentence, as shown in Table 8.

Table 8. Converting Stopwords to Sentences

Before Sentences	After Sentences
['sangka', 'kelas', 'menteri', 'kemensos', 'korupsi', 'bansos', 'dasar', 'biadab', 'adab', 'laknat']	sangka kelas menteri kemensos korupsi bansos dasar biadab adab laknat
['sentiasa', 'bangkit', 'peahankan', 'nabi', 'tuhan', 'kamiallah', 'agama', 'islam', 'islambangkit', 'tutupholywings', 'boikotnetflix', 'boikotindia', 'boikotchina', 'boikotacademyumbrella', 'shameonisrael']	sentiasa bangkit peahankan nabi tuhan kamiallah agama islam islambangkit tutupholywings boikotnetflix boikotindia boikotchina boikotacademyumbrella shameonisrael

f. Label Encoder

Then the results of preprocessing datasets are checked manually because some are null or blank. So, it is necessary to delete the data. Manual data are preprocessing yielded 3241 tweets, with HS labels on 2344 and non_HS labels on 897. The next step is to apply the encoder label. Encoder labels convert categorical or string data to numeric for easy modelling. Table 9 below shows the results of the HS label being 1 and the Non_HS label being 0.

Table 9. Labelling Encoders

Label	Tweet
1	sangka kelas menteri kemensos korupsi bansos dasar biadab adab laknat
0	sentiasa bangkit peahankan nabi tuhan kamiallah agama islam islambangkit tutupholywings boikotnetflix boikotindia boikotchina boikotacademyumbrella shameonisrael

3.2 FastText

FastText is an extension of Word2Vec [16], which uses the skip-gram method to find the most related words for a particular word. In research [6] using the skip-gram method with a window size of 5. This experiment was also carried out by research [20] using the Word2Vec skip-gram model with a window size of 15 modified by Text CNN on random oversampling, resulting in an F1-score of 93.70%. For this experiment, we used FastText from the gensim library, a free open-source Python library, to represent documents as semantic vectors. Gensim is designed to process raw and unstructured digital text ("plain text") using unsupervised algorithms [21]. In this research, we converted the tweet column into a list by tokenizing. Then, the FastText parameters used were vector_size=300, min_count=3, worker=4, epoch=1000, and skip-gram as experiments in this research. FastText will generate a class prediction for each given text example as its output. This output will display the class that the FastText model considers most probable for each text occurrence. In Table 10. below searches FastText for the same term for "biadab" and

Building of Informatics, Technology and Science (BITS)
Volume 4, No 3, Desember 2022 Page: 1548−1557
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v4i3.2718

"boikotchina." These results indicate that "biadab" is most similar to the word "adab." And the word that most closely resembles "boikotchina" is " boikotnetflix" in each of the 5 windows.

Table 10. Word Equation Search Results for FastText Embedding

Parameter	Similar Word	
Window	biadab	boikotchina
5	**('adab', 0.8036515712738037)**	**('boikotnetflix', 0.9899342656135559)**
10	('adab', 0.7701987028121948)	('boikotnetflix', 0.989216685295105)
15	('adab', 0.7756150960922241)	('boikotnetflix', 0.9879719018936157)

3.3 GloVe

This research will apply pre-trained word embedding GloVe by [17] because the model captures global corpus statistics directly. This experiment leverages corpus data from Twitter crawling that has been preprocessed by retrieving tweet attributes. To perform training, download the source code from [17]. We can convert text to numbers using GloVe. GloVe aspires to give vector representations with significant mathematical correlations between text words. Thus, we may execute mathematical operations on GloVe vectors to gain access to additional information regarding the meanings of words in the text. GloVe's output may determine the words most similar to a given the word by locating the vector most similar to the word vector. In the GloVe experiment, researchers employed parameters such as FastText, which differs from GloVe in that it does not use skipgrams. The researchers conducted a GloVe experiment to identify terms with a similar meaning to the target word. The researcher searched for the same term for "biadab" and "boikotchina" in Table 11. to identify the value with the closest similarity. Using window 5, the findings indicate that the closest term to "biadab" is "protes." Using window 15, the word "boikotchina" is equivalent to "boikotindia."

Tabel 11. Word Equation Search Results for GloVe Embedding

Parameter	Similar Word	
Window	biadab	boikotchina
5	**('protes', 0.9830155968666077)**	('bicara', 0.9335275292396545)
10	('koe', 0.9648107290267944)	('boikotindia', 0.9554847478866577)
15	('koe', 0.9571202993392944)	**('boikotindia', 0.9669917225837708)**

3.4 Random Oversampling

The HS dataset is unbalanced, with a minority belonging to the Non_HS class. This can negatively affect the performance of the trained model. Datasets also become unbalanced or called imbalanced datasets. Therefore, oversampling techniques are considered to achieve an unbiased model [22]. The type of data oversampling used equates data on the minority class with data on the majority class. The random oversampling method randomly selects tweets from the training dataset to be duplicated so that duplicated tweets are unknown [19]. Below is the result of random oversampling by adding the Non_HS class to Table 12.

Tabel 12. Before and After Results Using Random Oversampling Techniques

Class	Before Oversampling	After Oversampling
HS	2344	2344
Non_HS	897	2344
Total	**3241**	**4688**

3.5 LSTM

The LSTM model is used as a method for classification. LSTM will be compared to two embedded words, namely FastText and GloVe. Word embedding in 300-dimensional FastText and GloVe is trained with matrix embedding parameters obtained from the preprocessing feature of word embedding on the input. The LSTM architecture model used is Adam optimization, Binary Cross Entropy is used as a loss function, relu activation in a hidden gate, and sigmoid function activation for the output gate. Adam's optimization can handle the problem with infrequent gradients [23]. Then, using the dropout factor values suggested by [23], the researcher selected the dropout factor values of the medium model range [0.2, 0.35], specifically 0.2 and 0.4, for this experiment. If without dropout embedding, a higher probability of dropout in the repeating layer leads to overfitting, in contrast to the regularization of the embedding layer with the optimum range dropout probability [0.5], a higher probability of repeated layer dropout does lead to increased resistance to overfitting [23]. Binary Cross Entropy calculates the cross-entropy loss between the actual and predicted labels. Binary classification is selected if two labels are in the target set of values. The sigmoid activation function takes a value between 0 and 1. A value of <0.5 is set to 0, a value of >0.5 is set to 1, and evaluation of the model is performed by comparing it with the test data set [24]. Researchers conducted training in size 64 batches for word embedding and LSTM [10].

Building of Informatics, Technology and Science (BITS)
Volume 4, No 3, Desember 2022 Page: 1548−1557
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v4i3.2718

3.6 Experiment Results and Analyze

Researchers conducted two experimental scenarios. First, researchers compared the performance of FastText-LSTM with unbalanced and balanced data. Second, researchers compared the performance of GloVe-LSTM with unbalanced and balanced data. To evaluate this model, the dataset was divided into 80% each for training and 20% for testing [8]. The default scenario in this test is epoch 50, batch size 64. In addition, the researchers applied three window sizes, of which 5, 10, and 15, to observe the results [20]. Table 13. shows the classification results for each combination of LSTM with FastText, GloVe, and Random Oversampling. A higher F1-score was obtained at 89.57% when using a combination of FastText with LSTM, random oversampling methods, and window size 5.

Table 13. Experimental Results of FastText-LSTM, GloVe-LSTM with Three Different Window Sizes

Window Size	Method	Accuracy	Recall	Precision	F1-Score
5	FastText+LSTM	73.50%	79.76%	84.86%	82.23%
	FastText+LSTM+*Random Oversampling*	89.77%	91.35%	87.85%	**89.57%**
	GloVe+LSTM	71.19%	75.73%	88.49%	81.61%
	GloVe+LSTM+*Random Oversampling*	77.08%	77.61%	76.12%	76.86%
10	FastText+LSTM	71.34%	80.04%	80.38%	80.21%
	FastText+LSTM+*Random Oversampling*	89.45%	90.39%	88.27%	89.32%
	GloVe+LSTM	71.65%	75.40%	90.19%	82.14%
	GloVe+LSTM+*Random Oversampling*	74.41%	73.61%	76.12%	74.84%
15	FastText+LSTM	71.80%	81.64%	78.68%	80.13%
	FastText+LSTM+*Random Oversampling*	87.63%	91.92%	82.52%	86.97%
	GloVe+LSTM	72.88%	78.56%	85.93%	82.08%
	GloVe+LSTM+*Random Oversampling*	75.80%	78.81%	70.58%	74.47%

In Table 13. above the results on FastText-LSTM, if you use random oversampling, the accuracy, recall, precision, and F1-score are better than in FastText-LSTM. Because random oversampling is one of the regression techniques that can be used to solve the problem of data shortages in a dataset. Using random oversampling, we add duplicates of some instances in the dataset that are lacking, thereby increasing the number of instances in the underrepresented class. Random oversampling can help improve accuracy, recall, precision, and F1-scores in text classification using FastText-LSTM, as it increases the amount of data available for model training. With more data available, models can learn better from the data and produce more accurate predictions. The accuracy result of applying a random oversampling model to a FastText-LSTM model has a higher value than not applying a random oversampling model. Therefore, it can be concluded that the model studies duplicate data very well [19]. Meanwhile, GloVe-LSTM obtained the highest score with an F1-score value of 82.14%, in contrast to random oversampling, which gave an F1-score value of 76.86%. However, the experiment results had a good accuracy value of 77.08% compared to those obtained without random oversampling. In the case of GloVe-LSTM, random oversampling did not significantly improve classification results or even decrease them. This can be caused by overfitting, which occurs due to duplication of examples, and there need to be improvements to the model and training to improve classification results.

Based on the results of experiments in Table 13. FastText-LSTM models with random oversampling outperform GloVe-LSTM. The performance value results for windows 5 and 15 are not too far away and still have good performance values. Meanwhile, GloVe-LSTM with random oversampling needs to produce better performance values from GloVe-LSTM. In Table 14. The results of the confusion matrix comparisons for the FastText-LSTM-random oversampling-window 5 and the GloVe-LSTM-random oversampling-window 5. The FastText-LSTM-random oversampling-window 5 has accurate HS prediction results with 412 true positives, totaling 57 prediction errors. For Non_HS predictions, the model had accurate results with 430 true negatives and 39 prediction errors. The GloVe-LSTM-random oversampling-window 5 has accurate HS prediction results at 357 true positives and 112 false positive errors. For Non_HS prediction, the model had accurate results of 366 true negatives and 103 false negatives. The comparison of confusion matrix, accuracy, recall, precision, and f1-score in Table 14. below shows the experimental outcomes.

Table 14. Confusion Matrix Comparison Results on FastText-LSTM and GloVe-LSTM

	HS		Non_HS		Accuracy	Recall	Precision	F1-score
	TP	FP	TN	FN				
FastText+LSTM+Random Oversampling+Window 5	412	57	430	39	89.77%	91.35%	87.85%	89.57%
GloVe+LSTM+Random Oversampling+Window 5	357	112	366	103	77.08%	77.61%	76.12%	76.86%

In Table 15, we attest to predictions that focus on blasphemy, provocation, incitement, and explicit words. FastText-LSTM-random oversampling-window 5 has one example of provocation and incitement, namely 'kemensos targets act funds', with HS prediction results. GloVe-LSTM-random oversampling-window 5 was one of the

Building of Informatics, Technology and Science (BITS)
Volume 4, No 3, Desember 2022 Page: 1548−1557
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v4i3.2718

experiments whose predictions differed. On FastText, the prediction result for tweet number five is "Non_HS", whereas GloVe produces "HS". This proves that the accuracy value of GloVe could have been better compared to the FastText model. Table 15 below displays the prediction results.

Table 15. Text Prediction Results on FastText-LSTM and GloVe-LSTM

Tweet	FastText-LSTM-random oversampling-window 5	GloVe-LSTM-random oversampling-window 5
Cepat aksi kemensos duga langgar atur	0	0
Kominfo indihome bangsat	1	1
Tolol bodoh bajingan	0	0
Dana zakat dana sedekah infak maksimal	0	0
Gugat kemensos insyaallah izin terbit	0	1

4. CONCLUSION

In this research, a new dataset of tweets in Indonesian was created to identify hate speech. The dataset was manually labelled as containing hate speech or not and consisted of 3241 tweets, with 2344 in the "hate speech" class and 897 in the "non-hate speech" class. After using random oversampling to increase the size of the dataset to 4688 tweets, the best results were obtained using FastText with a window size of 5, which had an F1-score of 89.57% and an accuracy of 89.77%. In contrast, the GloVe model had an F1-score of 82.14% with a window size of 10 and no oversampling. The results showed that the FastText-LSTM embedding had a higher and more consistent rate of accurate classification compared to the GloVe-LSTM model. The researchers suggest that future research on identifying hate speech in Indonesian should carefully consider the proportion of the dataset and the type of hate speech identification used and should also consider tuning the parameters of the GloVe model to improve its performance.

REFERENCES

[1] "Digital in Indonesia: All the Statistics You Need in 2021," DataReportal – Global Digital Insights. https://datareportal.com/reports/digital-2021-indonesia (accessed Dec. 04, 2021).
[2] P. Fortuna and S. Nunes, "A Survey on Automatic Detection of Hate Speech in Text," ACM Comput. Surv., vol. 51, no. 4, p. 85:1-85:30, Jul. 2018, doi: 10.1145/3232676.
[3] M. Teja, "MEDIA SOSIAL: UJARAN KEBENCIAN DAN PERSEKUSI," p. 4.
[4] "Kasus Hate Speech Mendominasi Kejahatan Siber, Melebihi Laporan Konten Porno," kumparan. https://kumparan.com/kumparannews/kasus-hate-speech-mendominasi-kejahatan-siber-melebihi-laporan-konten-porno-1wEebgKLVuE (accessed Dec. 07, 2021).
[5] I. Alfina, R. Mulia, M. I. Fanany, and Y. Ekanata, "Hate Speech Detection in the Indonesian Language: A Dataset and Preliminary Study," Oct. 2017. doi: 10.1109/ICACSIS.2017.8355039.
[6] B. van Aken, J. Risch, R. Krestel, and A. Löser, "Challenges for Toxic Comment Classification: An In-Depth Error Analysis," ArXiv180907572 Cs, Sep. 2018, Accessed: Dec. 27, 2021. [Online]. Available: http://arxiv.org/abs/1809.07572
[7] T. Davidson, D. Warmsley, M. Macy, and I. Weber, "Automated Hate Speech Detection and the Problem of Offensive Language." arXiv, Mar. 11, 2017. Accessed: Jul. 25, 2022. [Online]. Available: http://arxiv.org/abs/1703.04009
[8] H. T.-T. Do, H. D. Huynh, K. Van Nguyen, N. L.-T. Nguyen, and A. G.-T. Nguyen, "Hate Speech Detection on Vietnamese Social Media Text using the Bidirectional-LSTM Model," ArXiv191103648 Cs, Nov. 2019, Accessed: Nov. 22, 2021. [Online]. Available: http://arxiv.org/abs/1911.03648
[9] T. Van Huynh, V. D. Nguyen, K. Van Nguyen, N. L.-T. Nguyen, and A. G.-T. Nguyen, "Hate Speech Detection on Vietnamese Social Media Text using the Bi-GRU-LSTM-CNN Model," ArXiv191103644 Cs, Dec. 2019, Accessed: Nov. 26, 2021. [Online]. Available: http://arxiv.org/abs/1911.03644
[10] P. Badjatiya, S. Gupta, M. Gupta, and V. Varma, "Deep Learning for Hate Speech Detection in Tweets," in Proceedings of the 26th International Conference on World Wide Web Companion - WWW '17 Companion, 2017, pp. 759–760. doi: 10.1145/3041021.3054223.
[11] A. Aprianto, "Reaksi Kebablasan Promosi Holywings," Tempo, Jun. 27, 2022. https://kolom.tempo.co/read/1606006/reaksi-kebablasan-promosi-holywings (accessed Jul. 25, 2022).
[12] T. detikcom, "Seruan Boikot Netflix dan The Umbrella Academy Gegara Lafaz Allah di Lantai," detikhot. https://hot.detik.com/tv-news/d-6147508/seruan-boikot-netflix-dan-the-umbrella-academy-gegara-lafaz-allah-di-lantai (accessed Jul. 25, 2022).
[13] "Kurang Ajar! Perusahaan China Jadikan Lafaz Allah Hiasan Bikini | Hukum." https://www.gatra.com/news-546669-hukum-kurang-ajar-perusahaan-china-jadikan-lafaz-allah-hiasan-bikini-.html (accessed Aug. 10, 2022).
[14] "'Kami Percaya ACT' Jadi Trending Topic, Publik Ramai Bandingkan dengan Tikus Berdasi," suara.com, Jul. 05, 2022. https://www.suara.com/news/2022/07/05/104130/kami-percaya-act-jadi-trending-topic-publik-ramai-bandingkan-dengan-tikus-berdasi (accessed Jul. 25, 2022).
[15] F. M. Sidik, "Ini Alasan Kemensos Cabut Izin Pengumpulan Uang dan Barang ACT," detiknews. https://news.detik.com/berita/d-6164336/ini-alasan-kemensos-cabut-izin-pengumpulan-uang-dan-barang-act (accessed Jul. 25, 2022).
[16] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching Word Vectors with Subword Information," ArXiv160704606 Cs, Jun. 2017, Accessed: Jan. 24, 2022. [Online]. Available: http://arxiv.org/abs/1607.04606

Building of Informatics, Technology and Science (BITS)
Volume 4, No 3, Desember 2022 Page: 1548−1557
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v4i3.2718

[17] J. Pennington, R. Socher, and C. Manning, "Glove: Global Vectors for Word Representation," in Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 2014, pp. 1532–1543. doi: 10.3115/v1/D14-1162.
[18] A. Bisht, A. Singh, H. Bhadauria, J. Virmani, and D. Kriti, "Detection of Hate Speech and Offensive Language in Twitter Data Using LSTM Model," 2020, pp. 243–264. doi: 10.1007/978-981-15-2740-1_17.
[19] C. Raj, A. Agarwal, G. Bharathy, B. Narayan, and M. Prasad, "Cyberbullying Detection: Hybrid Models Based on Machine Learning and Natural Language Processing Techniques," Electronics, vol. 10, no. 22, Art. no. 22, Jan. 2021, doi: 10.3390/electronics10222810.
[20] I. G. M. Putra and D. Nurjanah, "Hate Speech Detection In Indonesian Language Instagram," in 2020 International Conference on Advanced Computer Science and Information Systems (ICACSIS), Oct. 2020, pp. 413–420. doi: 10.1109/ICACSIS51025.2020.9263084.
[21] "What is Gensim? — gensim." https://radimrehurek.com/gensim/intro.html (accessed Aug. 14, 2022).
[22] M. F. Ahmed, Z. Mahmud, Z. T. Biash, A. A. N. Ryen, A. Hossain, and F. B. Ashraf, "Cyberbullying Detection Using Deep Neural Network from Social Media Comments in Bangla Language," ArXiv210604506 Cs, Jun. 2021, Accessed: Oct. 18, 2021. [Online]. Available: http://arxiv.org/abs/2106.04506
[23] W. K. Sari, D. P. Rini, and R. F. Malik, "Text Classification Using Long Short-Term Memory With GloVe Features," J. Ilm. Tek. Elektro Komput. Dan Inform., vol. 5, no. 2, Art. no. 2, Dec. 2019, doi: 10.26555/jiteki.v5i2.15021.
[24] "[1512.05287] A Theoretically Grounded Application of Dropout in Recurrent Neural Networks." https://arxiv.org/abs/1512.05287 (accessed Nov. 29, 2022).