COMPARISON OF GLOVE AND FASTTEXT ALGORITHMS ON CNN FOR CLASSIFICATION OF INDONESIAN NEWS CATEGORIES

[1]Tjong, Genesius Hartoko, [2]Hironimus Leong
[1,2]Department of Informatics Engineering, Faculty of Computer Science
Soegijapranata Catholic University
[2]marlon.leong@unika.ac.id

ABSTRACT

Computers cannot understand natural language as humans do, so natural language needs to be converted into something that computers can understand. Word embedding is a term that refers to a method for representing words in natural language into vectors so that computers can understand and perform mathematical operations. In a previous study, the classification of Indonesian news using CNN was carried out but only using the GloVe word embedding algorithm, while in another study it was found that fastText outperformed GloVe in terms of accuracy when classifying English news using CNN. However, because each language has different characteristics, grammar, and structure, this research was conducted to find out whether fastText would also outperform GloVe when using Indonesian news data. The dataset used in this study is a Wikipedia article to train the fastText and GloVe models which will produce a text representation in vector form and be used in the CNN model as a weight on the Embedding layer. The next dataset is Indonesian news with 8 categories for CNN model training, validation, and testing. This study will use 3 different numbers of Wikipedia articles to see the performance of each algorithm when given 10000, 50000, and 100000 Wikipedia articles. The results obtained from this study indicate that fastText outperforms GloVe in accuracy with an average difference of 2.51%, macro precision with an average difference of 4.32%, weighted precision with an average difference of 2.86%, and weighted recall with an average difference of 2.51 %, but for fastText macro recall it only excels when there are 10000 articles with a difference of 11.95% while when there are 50000 and 100000 articles GloVe excels with an average difference of 1.96%.

Keywords: glove, fasttext, indonesian news, cnn

INTRODUCTION

A computer can't understand natural language like a human does, so the natural language needs to be changed into something that the computer will understand. Word embedding is a term that refers to a method for representing words in natural language into vectors so that the computer can understand and perform a mathematical operation on it. Word embedding has an important role in natural language processing tasks like calculating distances between texts and also part-of-speech tagging, therefore word embedding has been developed until now.

The fact that the selection of the right word embedding method will have an impact on the performance of text classification and most natural language processing research uses deep artificial neural network is done using datasets in English, and there are rarely studies that use Indonesian language makes this research possible.

There are so many word embedding methods out there, but based on previous research, there are two methods that are quite promising to provide high accuracy, precision,

and recall. The two methods are GloVe and fastText.

In previous research, the classification of Indonesian news using CNN has been carried out but only using one type of word embedding method, namely GloVe and has not been carried out using other word embedding methods. In another study, it was shown that fastText outperforms GloVe in terms of accuracy for the classification of English news using CNN, but since each language has unique characteristics, grammar, and structure, it is necessary to review whether fastText will outperform GloVe using the Indonesian language news dataset. However, in other studies, it was shown that GloVe has higher accuracy for the classification of crisis tweets, but the algorithm used is a machine learning algorithm, namely Gaussian Naive Bayes, Random Forest, K Nearest Neighbors, and Support Vector Machines without trying more complex algorithms such as deep learning. Therefore, this study will compare the accuracy, precision, and recall of the CNN algorithm for classifying Indonesian news categories using fastText and GloVe as the word embedding methods.

RESEARCH METHODOLOGY

This chapter will explain the overall research methodology used. The research methodology is divided into 3 major parts, namely topic search, data processing, and observation of the algorithm as shown in Figure 1 and explained in detail in the next sub-chapter.

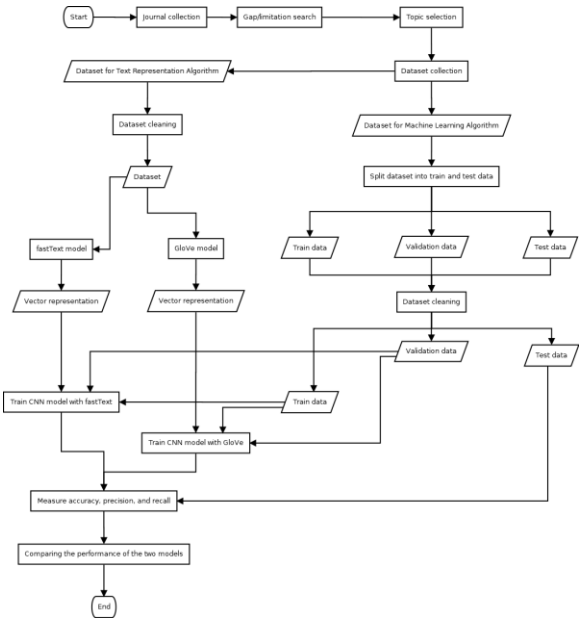

Figure 1. Research Methodology

Journal Collection

In order to find a gap/limitation in previous research as well as to find an algorithm that will be used in this study, it is necessary to search for journals that help prove the theory of this research. The journals used in this research originated from the internet which have been published to the general public and have almost the same theme, so they can support one another's findings.

After obtaining sufficient journals, a search for existing gaps/limitations is carried out based on these journals so that research topics can be selected, which will be research material in this report.

Dataset Collection

There are 2 data used in this study, the first data will be used to train the word embedding model, and the second data will be used to train the CNN model. Both data are sourced from:

1. The Wikimedia website entitled "idwiki dump progress on 20230420" [11] which contains data from the Indonesian language Wikipedia up to 20 April 2023 issued in XML or SQL form.
2. Mendeley's website entitled "Indonesian News Corpus" [12] which was published on 30 August 2018, contained 150466 news articles divided into eight categories consisting of economic business, football, lifestyle, national, sports, automotive, technology, and travels.

After collecting the necessary datasets, specifically for the second dataset, namely news articles, it is then divided into training data used to train the CNN model, validation data to see the quality of the model when viewing new data, and testing data to measure the performance of the CNN model based on accuracy, precision, and recall.

Dataset Cleaning

For the first data, data from the Wikimedia website will be cleaned by:

1. Change the existing formatting that is
    a. Omitting double square brackets (used to indicate that there is an explanation of the word on the Wikipedia website)
    b. Omit the first word between double square brackets separated by "|" (used as a marker that there is an explanation of the word on the Wikipedia website but with different keywords)
    c. Remove phrases between double square brackets that start with the word "*:" and end with "|" that appears first (used to insert files or categories)
    d. Eliminate phrases between square brackets that end with " " (used as a sign that there is an explanation of the word on a website other than Wikipedia)
    e. Eliminate two or more consecutive quotation marks (2 consecutive quotation marks are used as italics markers, 3 consecutive quotation marks as boldface)
    f. Removes the phrase between <ref> and </ref> (used to provide a reference)
    g. Eliminate two or more equal signs sequentially (used as sub-chapter markers)
    h. Remove the phrase between the curly braces (used to provide more information about a sub-chapter)
    i. Remove the phrases between {| and |}
    j. Remove the HTML and tags
2. Omitted Wikipedia pages that only contain redirects to other pages
3. Remove spaces before and at the end of sentences
4. Change to lowercase
5. Eliminate more than one consecutive space

For the second data, data from the Mendeley website which has been divided into train data, validation data, and test data will then be cleaned as below

1. Remove all special characters like "`", "!", "$", etc.
2. Eliminate spaces in front and end of sentences
3. Eliminate more than one consecutive space
4. Eliminate numbers
5. Change to lowercase
6. Eliminate words that include stopwords
7. Eliminate words that are less than 3 characters long

This is done to eliminate meaningless data and the effect is to reduce the dimensions of the data.

Word Embedding Algorithm

This study will compare two types of word embedding algorithms, namely fastText and GloVe.

fastText Model

The implementation of the fastText algorithm in this study was obtained from official sources [13], [14] using the C++ programming language.

To be able to do the training, the dataset is formatted with each article separated by a new line.

Sample documents:

Article 1 discusses the ...

Article 2 discusses the...

Article 3 discusses the ...

etc

After the data is formatted, the data is then inputted into the fastText model and training is carried out that produces a text representation in vector form. The vector will be inserted into the file with the format: word followed by the resulting vector, this is done with the aim that the vector can be loaded and used again at a later time.

Sample documents:

39605 300

and -0.14023109 0.16369423 -0.023652198 0.18757416 0.018635146 -0.4189826 0.38881835 -0.3090292 -0.106724195 -0.01472397 ... vector['and'][299]

the -0.048281312 0.12646523 0.033180393 -0.04841901 -0.023320947 -0.3154368 0.38430166 -0.18774219 -0.1372152 -0.012890713 ... vector['the'][299]

etc

The first line is the metadata, namely the number of words (39605) and the dimensions or size of the vector (300), the line after that is the word followed by the vector.

After the text representation in vector form is inserted into the file, the file is then loaded and processed in the form of a dictionary in the python programming language and used as the weight of the embedding layer for the CNN model.

GloVe Model

The implementation of the GloVe algorithm in this study was obtained from official sources [15] using the C programming language.

To be able to conduct training, the dataset input also needs to be formatted such as dataset input to Fasttext.

After the data is formatted, the data is inputted to the GloVe model and training is carried out and produces text representation in

the form of vectors. The resulting vectors will be inserted into files in the same format as fastText, this is done with the aim that the vectors can be loaded and used again at a later time.

After the representation of the text in the form of vectors is inserted into the file, the file is then loaded and processed into a dictionary form in the python programming language and is used as a weight of the embedding layer in the CNN model.

Training and Evaluation

The CNN model training process in this study used the Python programming language and used the help of the TensorFlow Library. There will be two types of training, namely CNN Model Training with fastText word embedding and with Glove word embedding, training will be conducted with the training dataset and will be evaluated with an evaluation dataset. After the training is completed, the CNN model will be used to predict the test data and the accuracy, precision, and recall will be calculated. In the end, the results will be compared between the two types of training, namely the CNN model with fastText word embedding and with GloVe word embedding.

RESULT AND DISCUSSION

Result

There are several results that can be seen in the table below. In the table below "accuracy" is the same as average "micro", so the columns are combined.

Table 1. Metrics from the CNN Model when Performing Tests

		Average		
Word Embedding - Amount of Data	**Metrics**	***Accuracy***	***Macro***	***Weighted***
fastText - 10000	Precision	0.866348	0.789237	0.873091
	Recall	0.866348	0.803771	0.866348
GloVe - 10000	Precision	0.810487	0.722756	0.807694
	Recall	0.810487	0.684277	0.810487
fastText - 50000	Precision	0.853554	0.793012	0.861060
	Recall	0.853554	0.746099	0.853554
GloVe - 50000	Precision	0.845047	0.763188	0.847454
	Recall	0.845047	0.771224	0.845047
fastText - 100000	Precision	0.859436	0.802750	0.859977
	Recall	0.859436	0.746058	0.859436
GloVe - 100000	Precision	0.84847	0.769587	0.853044
	Recall	0.84847	0.760157	0.848470

From table 1 when compared based on the number of datasets it can be seen that the accuracy of fastText is always higher than GloVe with an average difference of 2.51%. For macro precision, fastText is also always

higher with an average difference of 4.32%. For weighted precision fastText is also always higher with an average difference of 2.86%. For macro recall, when there is fewer data, namely 10000 articles, fastText gets a higher score with a difference of 11.95%, but when there is more data, namely 50000 and 100000 articles, GloVe gets a higher score with an average difference of 1.96%. For weighted recall, fastText is always higher with an average difference of 2.51%.

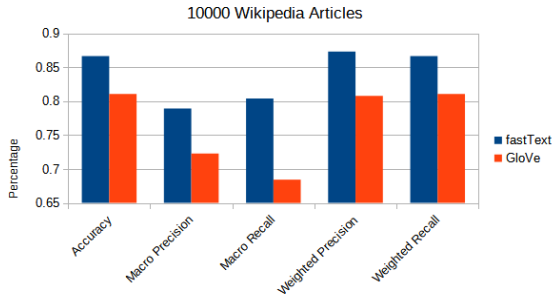

Figure 2. Comparison of fastText and GloVe with 10000 Wikipedia articles

Figure 2 shows a comparison of accuracy, macro precision, macro recall, weighted precision, and weighted recall between the prediction results of the CNN model using fastText and GloVe word embedding when trained with 10,000 Wikipedia articles. It appears that fastText scores higher than GloVe on all metrics.

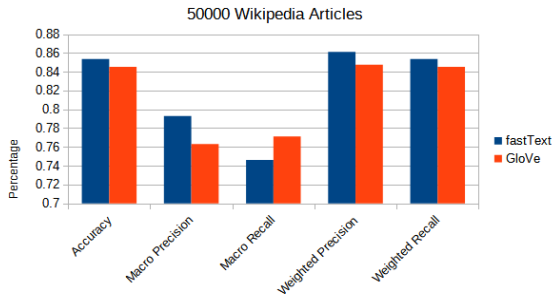

Figure 3. Comparison of fastText and GloVe with 50000 Wikipedia articles

Next is the comparison between the predicted results of the CNN model using fastText and GloVe word embedding when trained with 50,000 Wikipedia articles as shown in Figure 3. It can be seen that fastText scored higher than GloVe on all metrics except macro recall.

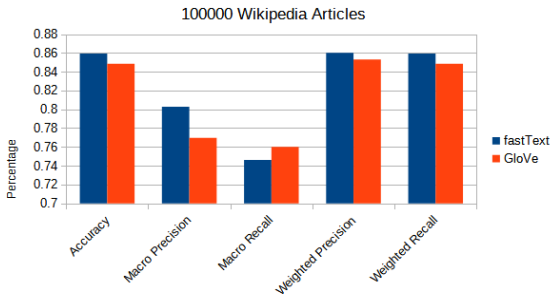

Figure 4. Comparison of fastText and GloVe with 100000 Wikipedia articles

For the final comparison between the predicted results of the CNN model using fastText and GloVe word embedding when trained with 100,000 Wikipedia articles, see Figure 4. It can be seen that fastText scored higher than GloVe on all metrics except macro recall.

Figure 5. Accuracy of the CNN Model when Conducting Tests

When compared to word embedding itself, it can be seen in Figure 5 that the accuracy of fastText decreased by 1.28% from 10000 to 50000 data but experienced an increase of 0.59% from 50000 to 100000 data but lower than 10000 data with a difference of 0.69%. So in fastText word embedding, 10000 data has the highest accuracy. Meanwhile, the accuracy of GloVe increased as more data increased, namely a 3.46% increase from 10000 to 50000 data and a 0.34% increase from 50000 to 100000 data. So in GloVe word embedding, 100000 data has the highest accuracy.

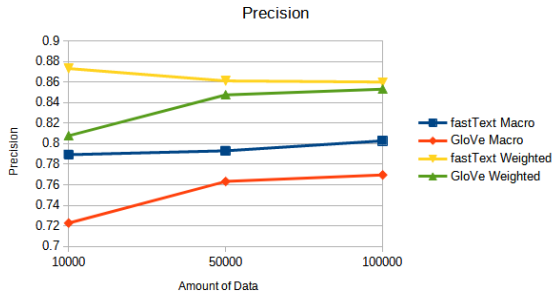

Figure 6. Precision of the CNN Model when Conducting Tests

For precision as shown in Figure 6. fastText macro precision increases with increasing data. FastText's weighted precision decreases as data increases. GloVe's macro precision increases with increasing data. And the weighted precision of GloVe also increases with increasing data.

Figure 7. Recall of the CNN Model when Conducting Tests

The last metric is recall as shown in Figure 7. The fastText macro recall decreases as data increases. Weighted recall fastText tends to be stable. GloVe macro recall tends to increase but slightly decreases from data 50000 to data 100000. And weighted recall GloVe increases as data increases.

Discussion

Based on the data described in the previous sub-chapter, it can be seen that fastText outperforms GloVe in almost every metric and every amount of data, although from the point of view of training time fastText is always longer than GloVe with a range of 39 to 84 percent longer which will certainly be different depending on the hardware specifications to train the two models. So the fastText algorithm is worth considering besides GloVe as a word embedding for Indonesian language news datasets.

The limitations of this study are that when training the fastText and GloVe models, the Wikipedia data is not randomized so that the data retrieved will be sorted alphabetically and there is also a news data gap that can reach 1:14.

CONCLUSION

Based on the test results of the 6 CNN models shown in Table 1, accuracy, precision, and also recall for each model and each word embedding when classifying Indonesian language news categories has quite satisfactory results with accuracy above 80%, weighted precision above 80%, and weighted recall above 80% although for macro precision and macro recall, it is in the range of 68 to 80

percent and this may occur due to gaps in the amount of data from each news category. For example, there is data that only has 605 samples but there are also those that have 12951 samples and if it is true that the cause of macro precision and macro recall is unsatisfactory due to the number of samples then this can be overcome by adding data for small samples.

And based on Table 1 it can also be seen that fastText almost always has higher accuracy, precision, and also recall than GloVe when classifying Indonesian language news categories, meaning that this research was successful in overcoming the limitations that existed in previous research where previous research only used GloVe word embedding.

Suggestions for further research are the use of more data for the fastText or GloVe models as well as improving the quality of the CNN model or even the use of other machine learning models which will result in higher accuracy, precision, and also recall.

REFERENCES

[1] Joulin, Armand, et al. "Bag of tricks for efficient text classification." arXiv preprint arXiv:1607.01759, https://arxiv.org/abs/1607.01759 (2016).

[2] Ramdhani, M. Ali, Dian Sa'adillah Maylawati, and Teddy Mantoro. "Indonesian news classification using convolutional neural network." Indonesian Journal of Electrical Engineering and Computer Science 19.2, https://pdfs.semanticscholar.org/e825/69350f83a20f88968d4035826bc529b8600a.pdf (2020): 1000-1009.

[3] Dharma, EDDY MUNTINA, et al. "The accuracy comparison among word2vec, glove, and fasttext towards convolution neural network (cnn) text classification." J Theor Appl Inf Technol 31.2, http://www.jatit.org/volumes/Vol100No2/5Vol100No2.pdf (2022).

[4] Li, Hongmin, et al. "Comparison of word embeddings and sentence encodings as generalized representations for crisis tweet classification tasks." Proceedings of ISCRAM Asia Pacific, https://par.nsf.gov/servlets/purl/10204524 (2018).

[5] Nguyen, Hai Ngoc, et al. "The Comparison of Word Embedding Techniques in RNNs for Vulnerability Detection." ICISSP, https://pdfs.semanticscholar.org/b0d0/772f51a98da5b2893bbbc1cc3f286c8f31c2.pdf (2021).

[6] Wang, Yanshan, et al. "A comparison of word embeddings for the biomedical natural language processing." Journal of biomedical informatics 87, https://www.sciencedirect.com/science/article/pii/S1532046418301825 (2018): 12-20.

[7] Pennington, Jeffrey, Richard Socher, and Christopher D. Manning. "Glove: Global vectors for word representation." Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), https://aclanthology.org/D14-1162.pdf (2014).

[8] Adipradana, Ryan, et al. "Hoax analyzer for Indonesian news using RNNs with fasttext and glove embeddings." Bulletin of Electrical Engineering and Informatics 10.4, https://www.beei.org/index.php/EEI/article/view/2956 (2021): 2130-2136.

[9] Keeling, Robert, et al. "Empirical comparisons of CNN with other learning algorithms for text classification in legal document review." 2019 IEEE International Conference on Big Data

(Big Data). IEEE, https://arxiv.org/pdf/1912.09499 (2019).
[10] David, Merlin Susan, and Shini Renjith. "Comparison of word embeddings in text classification based on RNN and CNN." IOP Conference Series: Materials Science and Engineering. Vol. 1187. No. 1. IOP Publishing, https://iopscience.iop.org/article/10.1088/1757-899X/1187/1/012029/meta (2021).
[11] idwiki dump progress on 20230420. https://dumps.wikimedia.org/idwiki/20230420/ (accessed May 2, 2023).
[12] Indonesian News Corpus. https://data.mendeley.com/datasets/2zpbjs22k3/1 (accessed March 13, 2023).
[13] Bojanowski, Piotr, et al. "Enriching word vectors with subword information." Transactions of the association for computational linguistics 5, https://arxiv.org/abs/1607.04606 (2017): 135-146.
[14] fastText source code. https://github.com/facebookresearch/fastText (accessed May 12, 2023).
[15] GloVe source code. https://github.com/stanfordnlp/GloVe (accessed May 15, 2023).