Jurnal Teknik Informatika (JUTIF)
Vol. 5, No. 1, February 2024, pp. 139-144

DOI: https://doi.org/10.52436/1.jutif.2024.5.1.1342
p-ISSN: 2723-3863
e-ISSN: 2723-3871

TOPIC CLASSIFICATION ON TWITTER USING CNN WITH WORD2VEC FEATURE EXPANSION

Rifaldy Bintang Ramadhan*[1], Erwin Budi Setiawan*[2]

[1,2]School of Computing, Universitas Telkom, Indonesia
Email: [1]bintangrifaldy@student.telkomuniversity.ac.id, [2]erwinbudisetiawan@telkomuniversity.ac.id

(Article received: August 21, 2023; Revision: September 30, 2023; published: February 08, 2024)

Abstract

Twitter is a social networking site that enables users to communicate with their followers by sending them short messages known as "tweets." Each tweet has a character limit of 280 characters. The minimum limit of tweets resulted in writing short tweets and increased use of word variations. This makes tweets difficult to understand without the help of the topic, thus tweets should be classified. This study aims to classify topics of Twitter using word2vec feature expansion to decrease vocabulary ambiguities in topic classification. This type of research is system design research. Feature expansion is a machine learning technique used to extract new features (or variables) from the dataset's existing features. A model's complexity and expressive power are intended to be increased through feature expansion in order to improve performance and generalization. Data were processed using Convolutional Neural Network (CNN). The results indicate that there is an important contribution in increasing understanding of topic classification in Twitter data with Word2Vec, and the CNN application is able to assist some obstacles in analyzing short text with high word variations.

Keywords: *CNN, feature expansion, topic classification, twitter, word2vec.*

1. INTRODUCTION

Twitter is a website owned and operated by Twitter Inc., which offers a social network in the form of a microblog that allows its users to post and read message Tweets [1]–[3]. Microblogging is an online communication tool where users can update statuses about those thinking and doing something and what they think about a particular object or phenomenon [4]. Tweets are written text of up to 280 characters displayed on the user's profile page [5]. Tweets are publicly viewable, but senders can limit message delivery to their friend lists. Users can view other users' Tweets, known as followers [6]–[8].

Natural Language Processing's text classification has numerous applications, including document classification, information search, sentiment analysis, and ranking [9]–[11]. The two main categories of the text classification model are: deep learning and machine learning Traditional machine learning algorithms, such as k-Nearest Neighbors, Naive Bayes, Support Vector Machine, and Logistic Regression that have been used in a lot of research on text classification [12].

Convolutional Neural network (CNN) is a deep learning method from the development of Multi-Layer Perceptron (MLP), which is designed to process two-dimensional data [13]. CNN is included in the type of Deep Neural network because it has a deep network level. CNN has two methods: classification using feed-forward and the learning stage using backpropagation [14]. The way CNN works is similar to MLP, but neurons are represented in two dimensions in CNN, whereas in MLP, each neuron is only one dimension in size. CNN was initially designed for image recognition but has developed into a versatile model for various tasks [1]. CNN can recognize local features in a multidimensional field. For example, in an image, CNN will find specific features, such as wheels or smiles, regardless of location [13].

This method will expand with the Word2vec feature, the word embedding model proposed by Mikolov, Sutskever, Chen, Corrado, and Dean in 2013 [4], [15]–[17]. Word2Vec can read and process text in large sizes and convert every word into a vector. This model can understand and identify the syntax and semantic meaning of words from natural languages and then represent each word with a vector [13]. Word2Vec achieves the best performance in NLP by similar grouping words that have the same vector [18].

The Neural Network is used in the Word2Vec model to produce output in the form of a vector space from the input in the form of a text corpus. The Word2Vec model has two types of architecture: Continuous Bag of Words (CBOW) and Skip-gram. The CBOW architecture predicts the current word based on context. In contrast, the Skip-gram architecture predicts within the range before or after the current word where the current word is input, so the Skip-gram architecture is considered an efficient architecture for studying large amounts of unstructured word vectors [9].

This study aims to classify topics on Twitter using word2vec feature expansion to reduce vocabulary ambiguities in topic classification. The practical benefit of this research is that this research can be used as material for consideration by Twitter users to reduce vocabulary ambiguity by clarifying topics using the Twitter tweet repository.

2. METHOD

The flowchart of the system design explanation can be seen in the following figure.

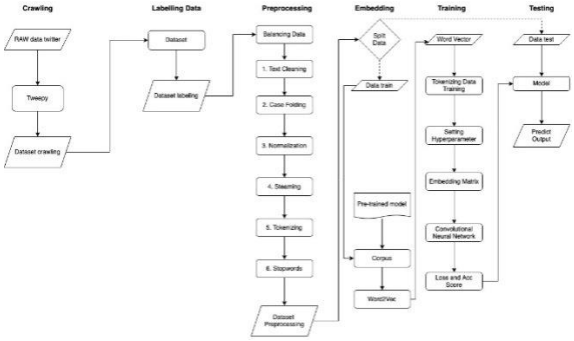

Figure 1. System Design Diagram

2.1. Data Collection

Data were obtained from the twitter.com website using the Twitter API with crawling methods and a tool called *tweepy* [19]. Collected data were used as training data and test data. Before conducting the categorization procedure, data labeling was carried out. data was labeled manually using Majority votes method.

Table 1. Example of labelling process

Tweet	Label
"kemarin nonton liverpool v chelsea terhibur karena banyak kepleset sama salah oper. Malem tadi nonton Arsenal v MU terhibur karena seru. Liga inggris memang penuh hiburan"	Hiburan
"Kerugian Ekonomi Akibat Gempa Dahsyat Turki-Suriah Diprediksi Rp60 Triliun"	Ekonomi

2.2. Data Preprocessing

Preprocessing data is the first step in processing raw data that will be entered into the classification system [13]. This process aims to prepare the data to be used efficiently in the classification system. In preprocessing data, the data balancing process was carried out first. This process aimed to overcome the unbalanced labelling class to ignore the minority class [12].

The data used is a post that is a user's text on the Twitter platform. If the data balancing process has been carried out, it can be continued data preprocessing with several stages as follows; 1) Text Cleaning, the process of removing noise is such as URLs, emojis, and symbols by reading the text thoroughly and then sentences containing noise are removed by deleting them.

2) Case folding, text data will be organized into a standard form to obtain the same format, converting words to lowercase throughout. 3) Normalization, At this stage, it was carried out to improve the spelling of abbreviated or extended words into common words by the Indonesian Dictionary *(KBBI)*. 4) Steaming**,** In this process, the data in these words were managed to find the words' basic by removing the suffix and prefix. 5) Tokenizing, In this stage, previously entire sentences were separated by detecting spaces in the sentence so that later the sentence can become a token form. 6) Stopwords, which aims to reduce the number of words in a document which can later affect the speed of NLP performance.

2.3. Embedding Data

After carrying out the data preprocessing stage, the next step is to divide the data into two, namely for training data and data testing. To share data, it was done by utilizing the library in python by using the function_train_test_split in the Sklearn library. The data embedding used the Word2vec model with the process of collecting tweets using *the skip gram*.

2.4. Word2Vec

Feature expansion enriches the original text by adding semantics to make it look like a large document [18]. In this study, the author used Word2vec to expand features. Word2vec refers to a group of models developed by Mikolov et al [20]. Word2Vec is used to create and train semantic vector spaces, often consisting of several hundred dimensions, based on a collection of text [9]. In this vector space, every word of the corpus is represented as a vector. Word2vec works on the basis that a word is only related to the words around it and has nothing to do with other words in the text. Word2vec's two main models are CBOW and skip-gram.[26].

$$L = \sum wt \in clogp(wt|Context(wt)) \quad (1)$$

$$L = \sum wt \in clogp(Context(wt)|wt) \quad (1)$$

Words that share context are geographically adjacent in that vector space—the Word2vec model by Mikolov et al. The study has received much attention in recent years. Vector representations of the words examined in the Word2vec model have semantic meanings and are helpful for various NLP tasks [21].

2.5. Convolutional Neural Network

The results of word vectorization that have been carried out in the embedding process are then carried out with the CNN method using two convolution layers, two polling layers and the addition of two batches of normalization, two dropouts and fully-connected layers. The stages in the CNN algorithm

are as follows; 1) get a feature map from the calculation results of the Convolutional Layer based on the predetermined filter size and form a matrix with a size of n x m x p; 2) do layering stage by taking the maximum value using the n x n filter from each feature map acquisition, so that the size of the matrix will be smaller than the convolutional matrix; and 3) perform batch normalization to speed up the training process in model creation by equalizing the distribution of each input value constantly changing due to parameter changes in the previous layer during the train model process.

4) After batch normalization, drop-out was carried out to reduce overfitting during the training process by randomly deleting either a hidden or a visible layer. 5) The next step is to reshape the feature map of a vector matrix into a 1 x n vector. 6) After getting the 1 x n vector, process the fully connected layer with n layers. 7) In the last stage, get the result by calculating the Softmax obtained from a fully connected layer.

3. RESULT AND DISCUSSION

3.1. Dataset

The dataset is obtained from Twitter which was crawled before then stored in a file and labeled using the opinions of several people (Majority Voting).

3.2. Libraries

The research used several popular python libraries which were used for scientific data processing, including Pandas, Numpy, Sklearn, Seaborn, and Matplotlib.

```
[ ] import pandas as pd
    import numpy as np
    import matplotlib.pyplot as plt
    import seaborn as sns
    import tensorflow as tf
    import warnings,re,string,nltk,re,gensim
    from sklearn.metrics import classification_report, confusion_matrix
    from nltk.corpus import stopwords
    from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
    #from tensorflow import keras
    from tensorflow.keras.models import Sequential, load_model
    from tensorflow.keras.preprocessing.text import Tokenizer
    from tensorflow.keras.preprocessing.sequence import pad_sequences
    from tensorflow.keras.layers import Embedding, Conv1D, GlobalMaxPooling1D, Dense, Dropout

    warnings.filterwarnings("ignore")
```

Figure 2. Libraries Used

3.3. Datasets Reading

After importing the libraries, the next step was to read the dataset that had been obtained using the Pandas library, with the script as follows:

```
[ ] df=pd.read_csv("dataset.csv")
    df=df[['Tweet','majority_votes']]
    df.rename(columns={"Tweet": "tweet","majority_votes": "topik"}, inplace=True)
    df.head()
```

	tweet	topik
0	3 Tips Perawatan Rem Cakram Motor GampangÂ Ko...	gaya hidup
1	Panduan Tips Pergi Liburan Ke Unggaran - https...	gaya hidup
2	Sefruit tips, coba jalan2 keluar cari takjil k...	gaya hidup
3	#dBsMorningTime Good Morning kawula muda..... ...	gaya hidup
4	9 Tips Terbaik Mengatasi Dehidrasi saat Berpua...	gaya hidup

Figure 3. Read Dataset

3.4. Convert from Categorical Data to Numerical

The next step was to convert categorical data into numerical, for example topic data is changed to: 0 for Lifestyle, 1 for Automotive, 2 for Sports, 3 for Education, and so on up to 9 for Economy.

```
[ ] topik=df['topik'].unique()
    print(topik)

    ['gaya hidup' 'otomotif' 'olahraga' 'edukasi' 'kesehatan' 'teknologi'
     'hiburan' 'bisnis' 'travel' 'ekonomi']

[ ]
    df['topik'].replace(topik,list(range(len(topik))), inplace=True)
    df.head()
```

	tweet	topik
0	tips perawatan rem cakram motor gampangâ kok	0
1	panduan tips pergi liburan ke unggaran	0
2	sefruit tips coba jalan keluar cari takjil kal...	0
3	dbsmorningtime good morning kawula muda bareng...	0
4	tips terbaik mengatasi dehidrasi saat berpuasa	0

Figure 4. Convert from Categorical Data to Numerical

3.5. Distribution of Datasets

The following is the distribution of the dataset based on its class, namely the stroke and normal classes:

```
(array([4926., 5743., 6046., 5900., 6045., 5121., 5535., 7244., 4027.,
        5410.]),
 array([0. , 0.9, 1.8, 2.7, 3.6, 4.5, 5.4, 6.3, 7.2, 8.1, 9. ]),
 <BarContainer object of 10 artists>)
```

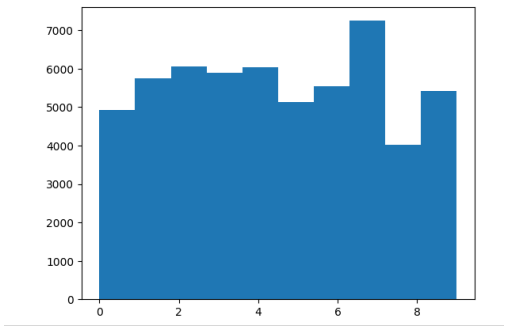

Figure 5. Graph Dataset

3.6. Building word2vec model

The above script is used to build the Word2Vec model, tokenize text, convert text into a token sequence, pad the token sequence, and build an embedding matrix based on the Word2Vec model that has been created. The following is the flow of the script:

```
[ ] # Membangun model Word2Vec
    sentences = preprocessed_df['tweet'].tolist()
    word2vec_model = gensim.models.Word2Vec(sentences, vector_size=100, min_count=1)
    word2vec_model.train(sentences, total_examples=word2vec_model.corpus_count, epochs=word2vec_model.epochs)

    # Membangun tokenizer
    tokenizer = Tokenizer()
    tokenizer.fit_on_texts(sentences)

    # Mengubah teks menjadi urutan token
    sequences = tokenizer.texts_to_sequences(sentences)

    # Padding urutan token menjadi panjang yang sama
    max_seq_length = max([len(seq) for seq in sequences])
    sequences = pad_sequences(sequences, maxlen=max_seq_length, padding='post')

    # Membangun matriks embedding menggunakan model Word2Vec
    embedding_matrix = np.zeros((len(word2vec_model.wv) + 1, word2vec_model.vector_size))
    for i, word in enumerate(word2vec_model.wv.index_to_key):
        embedding_matrix[i + 1] = word2vec_model.wv[word]
```

Figure 6. Building the Word2vec Model

1. Building the Word2Vec Model in the first part of the script. The steps for building the

Word2Vec model were performed, including: Retrieve text data from the 'tweet' column in preprocessed_df and store it in the form of list sentences.
Build the Word2Vec model with the following parameters:

- sentences: Text data that has been fetched before.
- vector_size: The number of vector dimensions a word represents.
- min_count: The minimum number of occurrences of a word to be included in the dictionary.

Conduct a Word2Vec model training with the following parameters:

- sentences: Text data that has been fetched before.
- total_examples: Total number of examples used in training (same as corpus_count from Word2Vec model).
- epochs: The number of training epochs to be performed.

2. Build Tokenizer:
   After building the Word2Vec model, the next part of the script builds a tokenizer. The tokenizer is used to convert text into a sequence of tokens or numbers that represent the words in the text.
3. Turning Text into Token Sequence:
   The next step is to convert the text (sentences) into token sequences using the previously created tokenizer. The results are stored in the form of list sequences.
4. Token Sequence Padding:
   Padding is used to make all sequences of tokens the same length because machine-learning models usually require dimensionally consistent input. In this script, the length of the token sequence is taken from the longest sequence. Then, the shorter token sequences were padded. Padding process was done by adding a zero value to the end of the token sequence (padding='post').
5. Building the Embedding Matrix:
   The final step was to build the embedding matrix using the Word2Vec model that was created earlier. The embedding matrix is a vector representation of the words in the corpus, where each row represents a vector representation of a single word. In this script:
   - The embedding matrix is initialized to zero and has the dimension (word_count + 1) x vector_size, where word_count is the number of words in the Word2Vec dictionary.
   - Loop through each word in the Word2Vec dictionary, and its vector representation (embedding) from the Word2Vec model is taken and inserted into the embedding matrix.

3.7. Data Splitting

Splitting data is a stage for dividing data into training data, and data testing implementation can be seen in the following figure:

```
# Split data menjadi data latih dan data uji (80% data latih, 20% data uji)
from sklearn.model_selection import train_test_split
train_data, test_data, train_labels, test_labels = train_test_split(sequences, preprocessed_df['topik'],
```

Figure 7. Splitting Data

3.8. CNN Training

Then use the Confusion Matrix to measure performance in binary classification problems and multiclass classification problems. The following is the implementation of the confusion matrix.

```
num_classes = len(preprocessed_df['topik'].unique())

# Membangun model CNN
model = Sequential()
model.add(Embedding(input_dim=embedding_matrix.shape[0], output_dim=embedding_matrix.shape[1], weights=[embedding_matrix],
model.add(Conv1D(128, 5, activation='relu'))
model.add(GlobalMaxPooling1D())
model.add(Dense(128, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(num_classes, activation='softmax'))

# Mengompilasi model
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])

# Melatih model dengan epoch 10
epochs = 25
history=model.fit(train_data, tf.keras.utils.to_categorical(train_labels, num_classes=len(topik)), epochs=epochs, validatio

# Evaluasi model pada data uji
loss, accuracy = model.evaluate(test_data, tf.keras.utils.to_categorical(test_labels,num_classes=len(topik)))
print('Loss:', loss)
print('Accuracy:', accuracy)
```

Figure 8. CNN Training

```
# Menghitung confusion matrix
cm = confusion_matrix(test_labels, predicted_labels)

# Membuat plot confusion matrix
plt.figure(figsize=(8, 6))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', cbar=False,xticklabels=topik, yticklabels=topik)
plt.title('Confusion Matrix')
plt.xlabel('Predicted Labels')
plt.ylabel('True Labels')
plt.show()
```

Figure 9.Confusion Matrix code

3.9. Analysis and Evaluation

From the performance results above, it can be seen that the precision value is 94%, the recall is 93%, the f1-score is 93%, and the support value is 93%. This shows the high accuracy and this model is suitable for use in topic detection or classification.

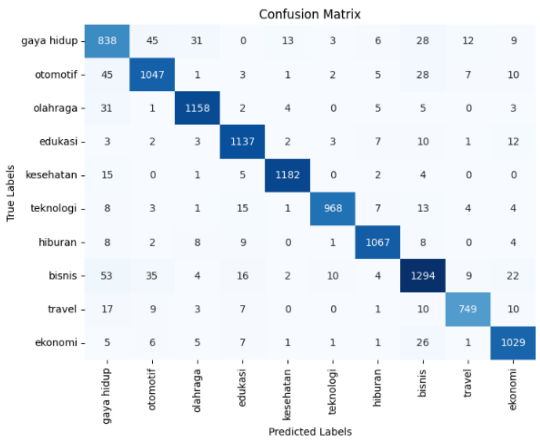

Figure 10. confusion matrix

Classification Report:

	precision	recall	f1-score	support
0	0.82	0.85	0.83	985
1	0.91	0.91	0.91	1149
2	0.95	0.96	0.96	1209
3	0.95	0.96	0.96	1180
4	0.98	0.98	0.98	1209
5	0.98	0.95	0.96	1024
6	0.97	0.96	0.96	1107
7	0.91	0.89	0.90	1449
8	0.96	0.93	0.94	806
9	0.93	0.95	0.94	1082
accuracy			0.93	11200
macro avg	0.94	0.93	0.93	11200
weighted avg	0.94	0.93	0.93	11200

Figure 11. Classification Report

Also the The training loss indicates how well the model is fitting the training data, while the validation loss indicates how well the model fits new data.

Figure 12. Training and Loss Validation

This research underscores the importance of topic classification in data from the Twitter platform. It also has an Application Program Interface (API) that enables integration between Twitter and other web applications and services. As one of the most extensive services, Twitter users have overgrown and attracted many companies' attention to customer habits. News organizations use Twitter to receive information about hazards and natural disasters. In addition, some businesses and organizations also use it to convey information to stakeholders. The limited number of characters in each tweet and the variety of words used by users make understanding the context of tweets difficult without topic classification. Therefore, the feature expansion technique using Word2Vec proved to be effective in overcoming ambiguity in vocabulary when classifying topics. Word2Vec allows words to be transformed into a numerical vector representation that takes their semantic meaning. The use of feature expansion techniques provides better complexity and expressive power to the model. In this way, the model's performance in classifying topics is improved, and its generalizability is also improved. Application of Convolutional Neural Network (CNN) to text data from Twitter, although originally developed for image processing, was successfully modified to carry out text classification. This indicates the flexibility and adaptability of the neural network model architecture. Achievements in this study sets a classification performance target with an accuracy rate of 93%.

4. CONCLUSION

The experimental results reveal that this target can be achieved, which shows that the developed model is able to classify topics in Twitter data properly. Overall, this research makes an important contribution in increasing understanding of topic classification in Twitter data. The use of feature expansion techniques with Word2Vec and the application of CNN helps overcome obstacles in the analysis of short texts with high word variations. By achieving the set performance targets, this study indicates the potential for using a similar model in a wider range of practical applications.

Future research is expected to expand the exploration of feature expansion techniques beside Word2Vec, such as FastText or GloVe to determine if there is a significant increase in topic classification performance. Moreover, create architectural variations of models other than CNNs, such as Recurrent Neural Networks (RNNs) or transformers which aims to recognize if there is an improvement in topic classification in Twitter text data. Then, collecting data by taking from different sources or different time periods to test the extent to which the developed model can generalize to more varied data.

REFERENCE

[1] F. A. R. Putra and Y. Sibaroni, "Detection of Radicalism Speech on Indonesian Tweet Using Convolutional Neural Network," *Build. Informatics, Technol. Sci.*, vol. 4, no. 2, 2022, doi: 10.47065/bits.v4i2.1907.

[2] S. A. El Rahman, F. A. Alotaibi, and W. A. Alshehri, "Sentiment Analysis of Twitter Data," *2019 Int. Conf. Comput. Inf. Sci. ICCIS 2019*, 2019, doi: 10.1109/ICCISci.2019.8716464.

[3] K. Chen, Z. Duan, and S. Yang, "Twitter as research data," *Polit. Life Sci.*, vol. 41, no. 1, pp. 114–130, 2022, doi: 10.1017/pls.2021.19.

[4] H. F. Naufal and E. B. Setiawan, "Ekspansi Fitur Pada Analisis Sentimen Twitter Dengan Pendekatan Metode Word2Vec," *e-Proceeding Eng.*, vol. 8, no. 5, pp. 10339–10349, 2021, [Online]. Available: https://dev.twitter.com

[5] R. Bharathi, R. Bhavani, and R. Priya, "Twitter Text Sentiment Analysis of Amazon Unlocked Mobile Reviews Using Supervised Learning Techniques," *Indian J. Comput. Sci. Eng.*, vol. 13, no. 4, pp. 1242–1253, 2022, doi: 10.21817/indjcse/2022/v13i4/221304100.

[6] E. M. Dharma, F. L. Gaol, H. L. H. S. Warnars, and B. Soewito, "the Accuracy Comparison Among Word2Vec, Glove, and Fasttext Towards Convolution Neural Network (Cnn) Text Classification," *J. Theor. Appl. Inf. Technol.*, vol. 100, no. 2, pp. 349–359, 2022.

[7] S. Rani and J. Singh, "Sentiment Analysis of Tweets Using Support Vector Machine," *Int. J. Comput. Sci. Mob. Appl.*, vol. 5, pp. 83–91, 2017, [Online]. Available: www.xyz.com

[8] X. Liu, B. Kar, C. Zhang, and D. M. Cochran, "Assessing relevance of tweets for risk communication," *Int. J. Digit. Earth*, vol. 12, no. 7, pp. 781–801, 2019, doi: 10.1080/17538947.2018.1480670.

[9] J. Brownlee, "Deep Learning for Natural Language Processing : Develop Deep Learning Models for Natural Language in Python," *Mach. Learn. Mastery*, p. 414, 2017, [Online]. Available: http://web.stanford.edu/class/cs224n/readings/cs224n-2019-notes06-NMT_seq2seq_attention.pdf

[10] B. Yu and C. T. Silva, "FlowSense: A natural language interface for visual data exploration within a dataflow system," *IEEE Trans. Vis. Comput. Graph.*, vol. 26, no. 1, pp. 1–11, 2020, doi: 10.1109/TVCG.2019.2934668.

[11] K. Kreimeyer *et al.*, "Natural language processing systems for capturing and standardizing unstructured clinical information: A systematic review," *J. Biomed. Inform.*, vol. 73, pp. 14–29, 2017, doi: 10.1016/j.jbi.2017.07.012.

[12] 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, p. 85, 2020, doi: 10.26555/jiteki.v5i2.15021.

[13] S. Aslan, "MF-CNN-BILSTM: A Deep Learning-Based Sentiment Analysis Approach and Topic Modeling of Tweets Related to the Ukraine-Russia Conflict," *SSRN Electron. J.*, 2022, doi: 10.2139/ssrn.4218398.

[14] M. Nazmi, A. Malisi, and E. B. Setiawan, "Ekspansi Fitur dengan Word2Vec pada Klasifikasi Topik dengan Metode Naive Bayes-Support Vector Machine di Twitter," *eProceedings …*, vol. 9, no. 1, pp. 67–78, 2022, [Online]. Available: https://openlibrarypublications.telkomuniversity.ac.id/index.php/engineering/article/view/17390%0Ahttps://openlibrarypublications.telkomuniversity.ac.id/index.php/engineering/article/view/17390/17100

[15] A. Jaffe, Y. Kluger, O. Lindenbaum, J. Patsenker, E. Peterfreund, and S. Steinerberger, "The Spectral Underpinning of word2vec," *Front. Appl. Math. Stat.*, vol. 6, 2020, doi: 10.3389/fams.2020.593406.

[16] G. Di Gennaro, A. Buonanno, and F. A. N. Palmieri, "Considerations about learning Word2Vec," *J. Supercomput.*, vol. 77, no. 11, pp. 12320–12335, 2021, doi: 10.1007/s11227-021-03743-2.

[17] M. Jaca-Madariaga, E. Zarrabeitia-Bilbao, R. M. Rio-Belver, and M. F. Moens, "Sentiment Analysis Model Using Word2vec, Bi-LSTM and Attention Mechanism," *Lect. Notes Data Eng. Commun. Technol.*, vol. 160, pp. 239–244, 2023, doi: 10.1007/978-3-031-27915-7_43.

[18] F. Ismayanti and E. B. Setiawan, "Deteksi Konten Hoax Berbahasa Indonesia Di Twitter Menggunakan Fitur Ekspansi Dengan Word2vec," *eProceedings …*, vol. 8, no. 5, pp. 10288–10300, 2021, [Online]. Available: https://openlibrarypublications.telkomuniversity.ac.id/index.php/engineering/article/view/15697%0Ahttps://openlibrarypublications.telkomuniversity.ac.id/index.php/engineering/article/view/15697/15410

[19] H. anggit Taba, "Analisis Sentimen Pemberlakuan Pembatasan Kegiatan Masyarakat Menggunakan Convolutional Neural Network," pp. 1–88, 2022.

[20] B. Desikan, "Natural Language Processing and Computational Linguistics: A practical guide," *Packt Publ. Ltd*, 2018, [Online]. Available: https://books.google.com/books?hl=en&lr=&id=48RiDwAAQBAJ&oi=fnd&pg=PP1&dq=Python%27s+natural+language+processing+(NLP)+libraries,+such+as+NLTK+and+spaCy,+are+utilized+to+perform+sentiment+analysis,+keyword+extraction,+and+topic+modeling+from+code+comments

[21] H. Juwiantho *et al.*, "Sentiment Analysis Twitter Bahasa Indonesia Berbasis Word2vec Menggunakan Deep Convolutional Neural Network," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 7, no. 1, pp. 181–188, 2018.