p-ISSN: 2252-3006
e-ISSN: 2685-2411

FastText and Bi-LSTM for Sentiment Analysis of Tinder Application Reviews

Anak Agung Mayra Candra Dyanggi[a1], I Wayan Agus Surya Darma[a2], Ni Nyoman Ayu J. Sastaparamitha[a3]
[a]Department of Informatics, Institut Bisnis dan Teknologi Indonesia, Indonesia
e-mail: [1]gungmayraa@gmail.com, [2]surya@instiki.ac.id, [3]ayu.sasta@instiki.ac.id

Abstrak

Saat ini teknologi mempengaruhi segala aspek bermasyarakat, salah satu inovasi dan kreatifitas di bidang teknologi adalah munculnya aplikasi kencan online. Aplikasi tersebut memudahkan penggunanya untuk mendapatkan pasangan sesuai dengan kriteria masing-masing. Aplikasi kencan online terpopuler adalah Tinder. Maraknya penggunakan aplikasi kencan online mengundang sentimen kontroversial di Tengah Masyarakat. Untuk mengatasi masalah tersebut, dibutuhkan sebuah analisis sentimen untuk mengetahui pendapat dan pandangan pengguna mengenai Tinder. Penelitian ini mengusulkan model fastText dan Bi-LSTM digunakan untuk mengetahui performa optimasi metode fastText dan Bi-LSTM dalam analisis sentimen dan membandingkan performanya dengan model fastText dan Bidirectional Encoder Representations from Transformers (BERT). Berdasarkan percobaan, fastText dan Bi-LSTM menghasilkan kinerja tertinggi pada skenario fold ke-4 dengan akurasi sebesar 88%, mengungguli model fastText dan BERT pada analisis sentimen dataset ulasan pengguna di aplikasi Tinder.

Kata kunci: *Analisis sentimen, Bi-LSTM, fastText, Tinder, Ulasan Pengguna*

Abstract

Nowadays technology affects all aspects of society, one of the innovations and creativity in the field of technology is the emergence of online dating application. The application makes it easy for users to find a partner according to their respective criteria. The most popular online dating app is Tinder. The rise of the use of online dating applications invites controversial sentiments in the community. In order to address this problem, a sentiment analysis is needed to find out users' opinions and views about Tinder. This study proposed the fastText and Bi-LSTM to analyze the public sentiment and compares the performance with the fastText and BERT models. Based on the experiment, the proposed model produced the highest performance in the 4th fold scenario with 88% accuracy, outperformed the fastText and BERT models on sentiment analysis of user review datasets in the Tinder application.

Keywords: *Sentiment Analysis, Bi-LSTM, fastText, Tinder, User Review*

1. Introduction

The development of technology today cannot be separated from social life. Along with the development of technology, the delivery of information and communication is fast and easy to obtain without distance barriers. One of the new phenomena due to digitalization is the emergence of online dating applications. Love relationships that are very close and cannot be separated from people's daily lives are used as a form of creativity made by technology activists and then packaged into a practical and attractive form. Dating apps can help users bring together people who have interests with their respective criteria. The most popular online dating app is Tinder, with over 100 million downloads on the Google Play Store by 2023.

Tinder is a Global Positioning System (GPS)-based matchmaking service application that facilitates communication between mutually interested users, making it possible to get matches. Tinder has a dating app concept that presents partner recommendations based on hobbies, beliefs, place of residence, and sexual orientation. According to Forbes magazine in 2023, the Tinder application was included in the 9 best dating apps of 2023 [1]. In this study, the

p-ISSN: 2252-3006
e-ISSN: 2685-2411

authors chose the Google Play Store platform to collect data on user reviews of the Tinder application.

Text classification is the process of classifying text based on words, phrases and other combinations to determine predefined categories. Text classification can find corpus similarities as well as pre-labeled groups [2]. This study will classify the sentiment of Tinder application users using two (2) categories, namely positive and negative. Sentiment analysis aims to analyze and explore the understanding of feelings, emotions, or sentiments contained in text or data. The magnitude of the influence and benefits of sentiment analysis has resulted in sentiment analysis-based study and applications growing rapidly, many companies are focusing on sentiment analysis services [3].

Sentiment analysis is a field of study that falls between various fields such as Data Mining, Natural Language Processing (NLP) and Machine Learning that focuses on extracting sentiment in a sentence. Natural Language Processing (NLP) is one of the computer science fields of Artificial Intelligence that specializes in language or linguistics [4]. Sentiment analysis tasks are used to clarify the polarity of text contained in documents, sentences or opinions. The purpose of sentiment analysis is to obtain sentiment information contained in a text related to an object or problem that occurs.

FastText is a word embedding model created by the Facebook team and released in 2016. FastText converts text into a continuous vector that can later be used in tasks that use any language. FastText has excellent performance in quickly modeling large-scale datasets. Bidirectional Long Short-Term Memory (Bi-LSTM) is a combination of deep learning methods consisting of two LSTM layers [5]. According to study [7] in comparing the meaning of sentences, Bi-LSTM is able to prove to have better performance [6]. Bi-LSTM is very suitable for handling semantic problems with the ability to recognize patterns in a sentence from each processed word.

This study proposes sentiment analysis of the Tinder application using the fastText and Bi-LSTM methods to determine the optimization performance of the fastText and Bi-LSTM methods in sentiment analysis and to find out the opinions of Indonesian people on the application and using K-Fold Cross Validation as a reference for evaluating the model that has been created.

2. Research Method / Proposed Method

In this study, FastText and Bi-LSTM for sentiment analysis begins with the process of scraping data on the Tinder application on the Google Play Store. The collected review data then through data pre-processing to clean the data from symbols or noise in the sentence. After pre-processing, data labeling is then carried out to facilitate sentiment classification. Sentiment classification is divided into two categories, i.e., positive (1) and negative (0). After labeling, word embeddings will be done using fastText and model training using Bi-LSTM.

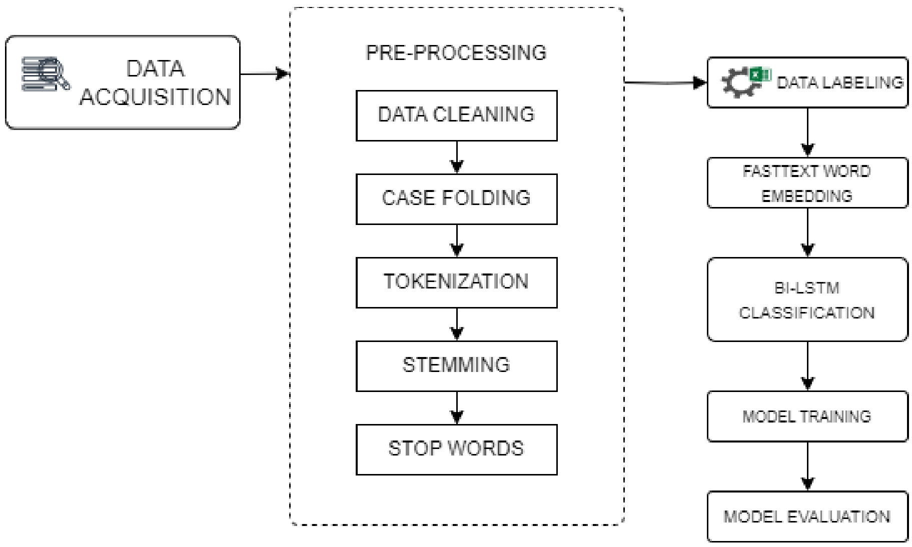

Figure 1. Proposed Method

 p-ISSN: 2252-3006
e-ISSN: 2685-2411

2.1. Data Acquisition

This study obtains datasets using the Google play scraper library which is used to retrieve data from the Google Play Store. Then use the pandas' library to manipulate and analyze the data. Google Play Scraper is a Python library that can retrieve data on Google Play Store application reviews. The Google Play Scraper library can filter the language or region you want to get data from, in this study the data taken is only Indonesian data. The data obtained from this study is 30,000.

2.2. Pre-Processing Data

Pre-processing is a process carried out to process unstructured data into a suitable format. In Figure 2, sentences are converted to lowercase and remove non-alphanumeric characters. The purpose of case folding is to convert all letters in the document into lowercase. Only letters 'a' to 'z' are accepted. Figure 3 shows the tokenizing process. Tokenizing is the process of separating text into pieces called tokens for later analysis. Words, numbers, symbols, punctuation marks, and other important entities can be considered as tokens.

Score	Content	Case Folding & Data Cleaning
1.0	Baru bikin akun lgsng di blokir. Wkwkwk lawak	baru bikin akun lgsng di blokir wkwkwk lawak
4.0	ok	ok
5.0	bagus	bagus
1.0	Banyak Akun Palsu, Akun Jual Diri, dan Profil yang menggunakan Tinder bukan untuk tujuan mencari pasangan	banyak akun palsu akun jual diri dan profil yang menggunakan tinder bukan untuk tujuan mencari pasangan
5.0	Good app	good app

Figure 2. Case Folding & Data Cleaning

Case Folding & Data Cleaning	Tokenized
keren banget	[keren, banget]
kenapa pas mau masuk login dari email trs gada pembritahuan dari email payah	[kenapa, pas, mau, masuk, login, dari, email, trs, gada, pembritahuan, dari, email, payah]
terlalu sering logout jadix bkib ulang barueh udah bkin ulang baru malah gk dpt2 jodoh	[terlalu, sering, logout, jadix, bkib, ulang, barueh, udah, bkin, ulang, baru, malah, gk, dpt2, jodoh]
kode verifikasi gk cocok apa cobak	[kode, verifikasi, gk, cocok, apa, cobak]
nice	[nice]

Figure 3. Tokenized

Figure 4 shows the stemming step to remove affixes or endings on words in a set of sentences to get the base word. The goal is to remove unnecessary word inflections, such as affixes and conjugations, so that words with the same root can be recognized as the same form. Figure 5 shows the stop words to reduce the dimensionality of the data and focus on more meaningful words that contribute to the content of the text. The main goal in applying this stopword process is to reduce the number of words in a document which will affect the speed and performance of NLP.

Data	Stemming
keren banget	keren banget
kenapa pas mau masuk login dari email trs gada pembritahuan dari email payah	kenapa pas mau masuk login dari email trs gada pembritahuan dari email payah
terlalu sering logout jadix bkib ulang barueh udah bkin ulang baru malah gk dpt2 jodoh	terlalu sering logout jadix bkib ulang barueh udah bkin ulang baru malah dpt2 jodoh
kode verifikasi gk cocok apa cobak	kode verifikasi cocok apa cobak
nice	nice

Figure 4. Stemming

p-ISSN: 2252-3006
e-ISSN: 2685-2411

Stemming	Stop Words
keren banget	keren banget
kenapa pas mau masuk login dari email trs gada pembritahuan dari email payah	pas masuk login email trs gada pembritahuan email payah
terlalu sering logout jadix bkib ulang barueh udah bkin ulang baru malah dpt2 jodoh	logout jadix bkib ulang barueh udah bkin ulang dpt2 jodoh
kode verifikasi cocok apa cobak	kode verifikasi cocok cobak
nice	nice

Figure 5. Stop Words

2.3. Labeling Data

In the data labeling stage, user comments or reviews will be labeled. Label determination is given according to the subjectivity of the researcher. The division of attribute classes in this study is divided into two, namely: positive and negative. Data labeling is an approach process called supervised learning. Labeling involves a linguist who labels positive and negative sentiments. Figure 6 shows the labeling of the dataset.

Tweet	Sentiment
mantab	positive
senang bahagia bgus	positive
apl kaga chatingan bayar nyesel gua download k...	negative
kecewa dgn aplikasi minggu match dgn wanita kt...	negative
cewenya sombong sombong banget gue nyari temen	negative

Figure 6. Labeling Data

2.4. Splitting Data

At this stage, we use K-Fold Cross Validation to split the data by 5-fold. The training data is used to train the machine to learn the patterns in the data. The validation data is used to evaluate the model during the training process. Data testing aims to test the performance of the model trained on the test data and measure how well the model predicts the data. Figure 7 shows the K-Fold Cross Validation in dataset.

Figure 7. K-Fold Cross Validation

3. Literature Study

Several studies have used fastText to perform sentiment analysis, in addition, fastText is usually combined with other methods. A study proposed by Aji *et al* compared the performance of word embedding using performance evaluation measured using F-measure. FastText is superior to two other word embedding methods with an F-measure value of 97.9% [8]. Fikry *et al* used a combination of FastText and SVM (Support Vector Machine). That study yielded good accuracy, which is 80% [9]. Malik and Sibaroni obtained an accuracy of 87.56% in using fastText and CNN for sentiment analysis of the TikTok application [10]. Husen *et al* used fastText and CNN in helping the selection of employee candidates resulting in a good accuracy of 84.01% [11]. Wibowo and Musdholifah used fastText and SVM, the experimental results show that the fastText-SVM model outperforms other models in terms of accuracy, which is 88.10% [12]. Alhazan *et al* using fastText for text classification showed that the proposed framework using fastText obtained a result of 89.9% [13]. Putra and Setiawan, using Feature Expansion with FastText Classification and Logistic Regression FastText scenario 90% training and 10% testing, obtained 71.8% [14]. Oshadi and Thelijjagoda, using fastText for sentiment analysis of user reviews on the Google Play Store obtained sentiment classification model accuracy results of 91%, with an f1 score of 85.97%, recall 85.93%, and precision 86.05%. The FastText model outperformed the Stanford CoreNLP library in the performance test [15]. Another study proposed by Zikri and Agustian resulted in an average accuracy of 82.65% using SVM and fastText for hate speech and abusive detection on Twitter [16].

Long Short-Term Memory (LSTM) is a Natural Language Processing model that is widely used for sentiment analysis tasks. Ouchene and Bessou proposed a model that combines FastText and (LSTM) showing its effectiveness with 88.95% accuracy [17]. Putri and Setiawan proposed the LSTM method and feature expansion with FastText that achieved an accuracy of 95.30% [18]. Furthermore, Bidirectional Long Short-Term Memory (Bi-LSTM) is a development of the LSTM model that can process words in 2 directions. Zhang and Guo used Bi-LSTM for sentiment analysis with the help of hyper-parameter adjustment and dropout mechanism, the experimental model evaluation indicator reached about 89% [19]. Nishide et al applied word2vec and bidirectional LSTM to train a model that aims at semantic correlation between forward and backward text, the experimental results show that the accuracy of emotion prediction is significantly improved [20]. Purwarianti and Crisdayanti proposed Bi-LSTM + WE (Word Embedding) model obtaining 91.66% precision, 91.26% recall, and 91.25% f-1 score [21]. Hu proposed online sentiment review research with Bi-LSTM on the test set, and the accuracy rate reached 89.03% [22].

Kiran *et al* proposed a study on social media trend analysis using Bi-LSTM resulting in a model that has the best performance with 98.72% accuracy, 93.65% precision, 94.02% recall, and f1-score 93.20% [23]. Thilaheswaran *et al*, used Bi-LSTM and GRU. The accuracy of bi-LSTM reached 98.65% [24]. Pramanik *et al*, analyzed sentiment with LSTM and Bi-LSTM. The resulting Bi-LSTM model has the highest accuracy of 97.25% compared to LSTM [25]. Tanriover O et al, produced the results of the ConvBiLSTM model with Word2Vec on the Tweet data set taken outperforming other models with 91.13% accuracy [26]. Mahara and Gangele, proposed research with a combination of RNN Bi-LSTM. The model obtained the best accuracy of 94% [27].

4. Result and Discussion

At this stage, the data will be trained in order to produce the best model. They are starting with word embeddings using fastText and classification with the Bidirectional Long Short-Term Memory (Bi-LSTM) model. Model evaluation using K-Fold Cross Validation Model analysis is performed by measuring classification based on train acc, train loss, val acc and val loss. K-Fold Cross Validation is a statistical method used to evaluate the performance of a model or algorithm that has been designed. In this study using Five-Fold Cross Validation. In 5-fold cross validation the data is divided into 5-folds of the same size if the total data owned is 25,000 then divided by 5, each fold totaling 5,000. Then, 4 folds will be used as training data and 1-fold is used as validation data. Furthermore, the model training process will take place 5 times with different subsets of training data and validation data at each iteration of the training process.

 p-ISSN: 2252-3006
e-ISSN: 2685-2411

4.1. Training Model

In this section, we presented the results of training model based on K-Fold Cross Validation. In each each fold, the data will be trained in order to produce the best model. The training process are starting with word embeddings using fastText and classification using the Bidirectional Long Short-Term Memory (Bi-LSTM) model.

Table 1. Training 1-Fold Cross Validation

Epochs	Train Acc	Train Loss	Val Acc	Val Loss
1	0.7034	0.6295	0.7606	0.5085
2	0.8147	0.4224	0.8373	0.3763
3	0.8601	0.3433	0.8552	0.3462
4	0.8811	0.3029	0.8617	0.3345
5	0.8945	0.2705	0.8648	0.3251
6	0.9079	0.2431	0.8665	0.3294
7	0.9211	0.2192	0.8667	0.3373
8	0.9335	0.1970	0.8640	0.3509
9	0.9390	0.1790	0.8650	0.3623
10	0.9451	0.1670	0.8623	0.4189
11	0.9491	0.1547	0.8531	0.4326
12	0.9527	0.1440	0.8587	0.4058
13	0.9564	0.1352	0.8579	0.4343
14	0.9580	0.1286	0.8577	0.4577
15	0.9577	0.1262	0.8594	0.4620

Table 1 shows the training model 1-fold cross validation produces the best training metrics values 'train loss: (0.1262, epochs 15), 'train_acc': (0.958, epochs 14). Best validation metrics value 'val loss': (0.3251, epochs 5), 'val_acc': (0.8667, epochs 7).

Table 2. Training 2-Fold Cross Validation

Epochs	Train Acc	Train Loss	Val Acc	Val Loss
1	0.6426	0.6395	0.7535	0.5294
2	0.8174	0.4254	0.8371	0.3949
3	0.8609	0.346	0.8502	0.3640
4	0.8808	0.3053	0.8604	0.3498
5	0.8928	0.2743	0.8602	0.3447
6	0.9102	0.2445	0.8648	0.3449
7	0.9220	0.2201	0.8644	0.3570
8	0.9349	0.1964	0.8617	0.3887
9	0.9433	0.1759	0.859	0.4026
10	0.9476	0.1618	0.8575	0.4084
11	0.9530	0.1483	0.856	0.4493
12	0.9561	0.1396	0.8529	0.4444
13	0.9580	0.1332	0.8527	0.4831
14	0.9588	0.1271	0.8521	0.4682
15	0.9601	0.1212	0.854	0.5209

Table 2 shows the training model 2-fold cross validation produces the best training metrics values 'train loss': (0.1212, epochs 15), 'train_acc': (0.9601, epochs 15). The best training validation values 'val loss': (0.3447, epochs 5), 'val_acc': (0.8648, epochs 6). In Table 3. Training model 3-fold cross validation produces the best training metrics values 'train loss': (0.1186, epochs 16), 'train_acc': (0.9597, epochs 16). The best training validation values 'val loss': (0.308, epochs 6), 'val_acc': (0.876, epochs 7). In Table 4. The 4-fold cross validation training model resulted in the best training metric values 'train loss': (0.1324, epochs 15), 'train_acc': (0.9553, epochs 14). Best training validation value 'val loss': (0.3089, epochs 5), 'val_acc': (0.88, epochs 7).

p-ISSN: 2252-3006
e-ISSN: 2685-2411

Table 3. Training 3-Fold Cross Validation

Epochs	Train Acc	Train Loss	Val Acc	Val Loss
1	0.6343	0.6419	0.7631	0.5140
2	0.8085	0.4341	0.8388	0.3755
3	0.8573	0.3528	0.8610	0.3359
4	0.8771	0.3109	0.8667	0.3181
5	0.8910	0.2795	0.8702	0.3173
6	0.9035	0.2540	0.8740	0.3080
7	0.9205	0.2264	0.8760	0.3207
8	0.9299	0.2050	0.8717	0.3280
9	0.9371	0.1859	0.8710	0.3408
10	0.9430	0.1686	0.8694	0.3598
11	0.9455	0.1597	0.8662	0.3631
12	0.9524	0.1451	0.8660	0.4108
13	0.9538	0.1375	0.8633	0.4250
14	0.9552	0.1315	0.8625	0.4344
15	0.9579	0.1249	0.8627	0.4589
16	0.9597	0.1186	0.8635	0.4779

Table 4. Training 4-Fold Cross Validation

Epochs	Train Acc	Train Loss	Val Acc	Val Loss
1	0.6483	0.6404	0.7602	0.5128
2	0.8112	0.4308	0.850	0.3750
3	0.8565	0.3500	0.8631	0.3401
4	0.8777	0.3102	0.8696	0.3200
5	0.8906	0.2792	0.8727	0.3089
6	0.9032	0.2527	0.8731	0.3104
7	0.9178	0.2280	0.880	0.3143
8	0.9306	0.2058	0.8681	0.3508
9	0.9367	0.1881	0.8763	0.3389
10	0.9425	0.1705	0.870	0.3538
11	0.9467	0.1570	0.869	0.3707
12	0.9507	0.1505	0.8679	0.3858
13	0.9533	0.1405	0.8633	0.4359
14	0.9553	0.1337	0.8654	0.4476
15	0.9545	0.1324	0.8669	0.4558

Table 5. Training 5-Fold Cross Validation

Epochs	Train Acc	Train Loss	Val Acc	Val Loss
1	0.6382	0.6392	0.7571	0.5168
2	0.8159	0.4274	0.8385	0.3859
3	0.8573	0.3462	0.8546	0.3549
4	0.8810	0.3045	0.8600	0.3399
5	0.8947	0.2736	0.8656	0.3319
6	0.9090	0.2434	0.8633	0.3416
7	0.9226	0.2196	0.8635	0.3566
8	0.9331	0.1983	0.8637	0.3597
9	0.9408	0.1786	0.8652	0.3810
10	0.9467	0.1624	0.8575	0.4275
11	0.9511	0.1494	0.8608	0.4296
12	0.9554	0.1387	0.8625	0.4314
13	0.9565	0.1343	0.8612	0.4472
14	0.9578	0.1293	0.8548	0.4711
15	0.9598	0.1221	0.8529	0.5077

Table 5 shows the 5-fold cross validation training model resulted in the best training metric values 'train loss': (0.1221, epochs 15), 'train_acc': (0.9598, epochs 15). Best training validation values 'val loss': (0.3319, epochs 5), 'val_acc': (0.8656, epochs 5). Figure 8 shows the

 p-ISSN: 2252-3006
e-ISSN: 2685-2411

accuracy validation graph of each fold, the fold that has the lowest accuracy is fold 2 (0.8648) and the highest is fold 4 (0.880).

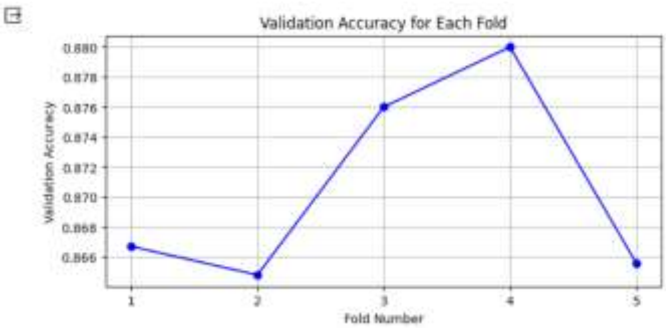

Figure 8. Plot Validation Accuracy for Each Fold

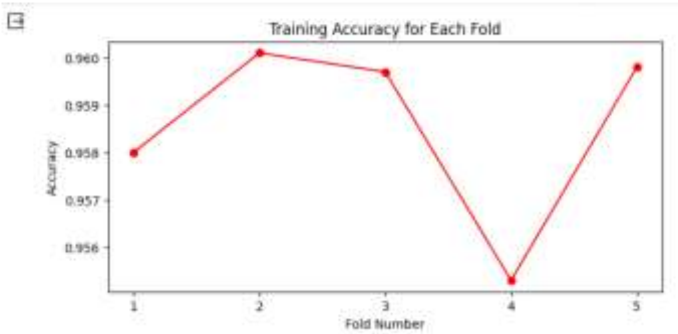

Figure 9. Plot Training Accuracy for Each Fold

Figure 9 shows the visualization results of the training accuracy plot for each fold, in these results it can be concluded that the 4th fold has the lowest accuracy result of 0.9553 and the highest is the 2nd fold of 0.9601. Table 6. shows that the sentiment model with the Tinder dataset with the highest accuracy is obtained from the fastText and BI-LSTM models. While the lowest is obtained from the fastText model.

Table 6. Model Performance Comparison

No	Model	Accuracy
1	fastText	86%
2	BERT	87%
3	fastText+ BI-LSTM	88%

a b

Figure 10. Word Cloud. (a) Positive Sentiments, (b) Negative Sentiments

Figure 10 shows specific words that often appear in positive Tinder sentiments, namely "good, good, not bad, ok, great, cool, helpful" etc. Comments mostly state that Tinder is useful for its users. Whereas in negative Tinder sentiments, specific words appear, namely "pay, paid, verify", "block", etc. Comments on negative Tinder sentiments mostly state that the app has

p-ISSN: 2252-3006
e-ISSN: 2685-2411

disadvantages for users such as blocked accounts and not being able to receive verification codes.

5. Conclusion

Based on the experimental results of this study, the following conclusions are drawn. We analyzed public sentiment towards the Tinder app by collecting user review data on the Google Play Store. The sentiment analysis model based on fastText and BI-LSTM produces the highest performance in the 4th fold scenario with an accuracy of 88%. Based on the comparison of the three model performances, the fastText and BI-LSTM models can outperform the fastText and BERT models on sentiment analysis of user review datasets in the Tinder application.

References

[1] C. P. Anna Baluch, “9 Best Dating Apps Of 2023: Reviews and Costs,” 2023.

[2] F. Handayani, D. Feddy, and S. Pribadi, “Implementasi Algoritma Naive Bayes Classifier dalam Pengklasifikasian Teks Otomatis Pengaduan dan Pelaporan Masyarakat melalui Layanan Call Center 110,” 2015. [Online]. Available: https://doi.org/10.15294/jte.v7i1.8585

[3] N. Herlinawati, Y. Yuliani, S. Faizah, W. Gata, and S. Samudi, “Analisis Sentimen Zoom Cloud Meetings di Play Store Menggunakan Naïve Bayes dan Support Vector Machine,” *CESS (Journal of Computer Engineering, System and Science)*, vol. 5, no. 2, p. 293, Jul. 2020, doi: 10.24114/cess.v5i2.18186.

[4] Y. Yunefri and Y. Ersan Fadrial, “CHATBOT PADA SMART COOPERATIVE ORIENTED PROBLEM MENGGUNAKAN NATURAL LANGUAGE PROCESSING DAN NAIVE BAYES CLASSIFIER CHATBOT ON SMART COOPERATIVE ORIENTED PROBLEMS USING NATURAL LANGUAGE PROCESSING AND NAIVE BAYES CLASSIFIER,” *Journal of Information Technology and Computer Science (INTECOMS)*, vol. 4, no. 2, p. 2021, 2021.

[5] D. I. Af’idah, D. Dairoh, S. F. Handayani, R. W. Pratiwi, and S. I. Sari, “Sentimen Ulasan Destinasi Wisata Pulau Bali Menggunakan Bidirectional Long Short Term Memory,” *MATRIK : Jurnal Manajemen, Teknik Informatika dan Rekayasa Komputer*, vol. 21, no. 3, pp. 607–618, Jul. 2022, doi: 10.30812/matrik.v21i3.1402.

[6] I. Habernal and I. Gurevych, “Which argument is more convincing? Analyzing and predicting convincingness of Web arguments using bidirectional LSTM,” 2016. [Online]. Available: https://github.com/UKPLab/

[7] Y. V. Aritonang, D. P. Napitupulu, M. H. Sinaga, and J. Amalia, “Pengaruh Hyperparameter pada Fasttext terhadap Performa Model Deteksi Sarkasme Berbasis Bi-LSTM,” *JATISI (Jurnal Teknik Informatika dan Sistem Informasi)*, vol. 9, no. 3, pp. 2612–2625, Sep. 2022, doi: 10.35957/jatisi.v9i3.1331.

[8] A. Nurdin, B. Anggo, S. Aji, A. Bustamin, and Z. Abidin, “PERBANDINGAN KINERJA WORD EMBEDDING WORD2VEC, GLOVE, DAN FASTTEXT PADA KLASIFIKASI TEKS,” *Jurnal TEKNOKOMPAK*, vol. 14, no. 2, p. 74, 2020.

[9] E. Budianita, M. Fikry, and F. Yanto, “Klasifikasi Sentiment Review Aplikasi MyPertamina Menggunakan Word Embedding FastText dan SVM (Support Vector Machine),” *Jurnal Sistem Komputer dan Informatika (JSON) Hal: 526−*, vol. 534, no. 3, 2023, doi: 10.30865/json.v4i3.5695.

[10] R. A. A. Malik and Y. Sibaroni, “Multi-aspect Sentiment Analysis of Tiktok Application Usage Using FasText Feature Expansion and CNN Method,” *Journal of Computer System and Informatics (JoSYC)*, vol. 3, no. 4, pp. 277–285, Sep. 2022, doi: 10.47065/josyc.v3i4.2033.

[11] F. Alfariqi, W. Maharani, and J. H. Husen, “Klasifikasi Sentimen pada Twitter dalam Membantu Pemilihan Kandidat Karyawan dengan Menggunakan Convolutional Neural Network dan Fasttext Embeddings,” 2020. [Online]. Available: https://www.alexa.com/siteinfo/twitter.com

[12] D. A. Wibowo and A. Musdholifah, “Sentiments Analysis of Indonesian Tweet About Covid-19 Vaccine Using Support Vector Machine and Fasttext Embedding,” in *2021 4th International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*, IEEE, Dec. 2021, pp. 184–188. doi: 10.1109/ISRITI54043.2021.9702871.

[13] A. Alessa, M. Faezipour, and Z. Alhassan, "Text Classification of Flu-Related Tweets Using FastText with Sentiment and Keyword Features," in *2018 IEEE International Conference on Healthcare Informatics (ICHI)*, IEEE, Jun. 2018, pp. 366–367. doi: 10.1109/ICHI.2018.00058.

[14] M. F. D. Putra and E. B. Setiawan, "Influence of Sentiment on Mandiri Bank Stocks (BMRI) Using Feature Expansion with FastText and Logistic Regression Classification," in *2022 International Conference on Advanced Creative Networks and Intelligent Systems (ICACNIS)*, IEEE, Nov. 2022, pp. 1–7. doi: 10.1109/ICACNIS57039.2022.10055450.

[15] D. M. K. Oshadi and S. Thelijjagoda, "AppGuider: Feature Comparison System using Neural Network with FastText and Aspect-based Sentiment Analysis on Play Store User Reviews," in *2022 3rd International Conference on Smart Electronics and Communication (ICOSEC)*, IEEE, Oct. 2022, pp. 1148–1155. doi: 10.1109/ICOSEC54921.2022.9952093.

[16] A. Zikri, A. Zikri, and S. Agustian, "Penerapan Support Vector Machine dan FastText untuk Mendeteksi Hate Speech dan Abusive pada Twitter," vol. 7, no. 1, pp. 436–443, 2023, doi: 10.30865/mib.v7i1.5408.

[17] L. Ouchene and S. Bessou, "FastText Embedding and LSTM for Sentiment Analysis: An Empirical Study on Algerian Tweets," in *2023 International Conference on Information Technology (ICIT)*, IEEE, Aug. 2023, pp. 51–55. doi: 10.1109/ICIT58056.2023.10226060.

[18] B. A. Putri and E. B. Setiawan, "Topic Classification Using the Long Short-Term Memory (LSTM) Method with FastText Feature Expansion on Twitter," in *2023 International Conference on Data Science and Its Applications (ICoDSA)*, IEEE, Aug. 2023, pp. 18–23. doi: 10.1109/ICoDSA58501.2023.10277033.

[19] T. Zhang and J. Guo, "Research on Chinese Sentiment Analysis Based on Bi-LSTM Networks," in *2021 IEEE/ACIS 20th International Fall Conference on Computer and Information Science (ICIS Fall)*, IEEE, Oct. 2021, pp. 198–203. doi: 10.1109/ICISFall51598.2021.9627460.

[20] J. Hu, X. Kang, S. Nishide, and F. Ren, "Text multi-label sentiment analysis based on Bi-LSTM," in *2019 IEEE 6th International Conference on Cloud Computing and Intelligence Systems (CCIS)*, IEEE, Dec. 2019, pp. 16–20. doi: 10.1109/CCIS48116.2019.9073727.

[21] A. Purwarianti and I. A. P. A. Crisdayanti, "Improving Bi-LSTM Performance for Indonesian Sentiment Analysis Using Paragraph Vector," in *2019 International Conference of Advanced Informatics: Concepts, Theory and Applications (ICAICTA)*, IEEE, Sep. 2019, pp. 1–5. doi: 10.1109/ICAICTA.2019.8904199.

[22] J. Hu, "Sentiment Classification Model of Online Reviews Based on Word Features and Bi-LSTM," in *2022 IEEE 2nd International Conference on Data Science and Computer Application (ICDSCA)*, IEEE, Oct. 2022, pp. 1101–1105. doi: 10.1109/ICDSCA56264.2022.9988320.

[23] M. Ati, M. U. G. Khan, and I. Kiran, "Social Media Trends Analysis using the Bi-LSTM with Multi-Head Attention," in *2022 International Conference on Electrical and Computing Technologies and Applications (ICECTA)*, IEEE, Nov. 2022, pp. 295–299. doi: 10.1109/ICECTA57148.2022.9990328.

[24] K. Mouthami, N. Yuvaraj, K. K. Thilaheswaran, and K. J. Lokeshvar, "Text Sentiment Analysis of Film Reviews Using Bi-LSTM and GRU," in *2023 4th International Conference on Electronics and Sustainable Communication Systems (ICESC)*, IEEE, Jul. 2023, pp. 1379–1386. doi: 10.1109/ICESC57686.2023.10193121.

[25] N. Kapali, T. Tuhin, A. Pramanik, Md. S. Rahman, and S. R. H. Noori, "Sentiment Analysis of Facebook and YouTube Bengali Comments Using LSTM and Bi-LSTM," in *2022 13th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, IEEE, Oct. 2022, pp. 1–6. doi: 10.1109/ICCCNT54827.2022.9984395.

[26] S. Tam, R. Ben Said, and O. O. Tanriover, "A ConvBiLSTM Deep Learning Model-Based Approach for Twitter Sentiment Classification," *IEEE Access*, vol. 9, pp. 41283–41293, 2021, doi: 10.1109/ACCESS.2021.3064830.

[27] G. S. Mahara and S. Gangele, "Fake news detection: A RNN-LSTM, Bi-LSTM based deep learning approach," in *2022 IEEE 1st International Conference on Data, Decision

p-ISSN: 2252-3006
e-ISSN: 2685-2411

and Systems (ICDDS), IEEE, Dec. 2022, pp. 01–06. doi: 10.1109/ICDDS56399.2022.10037403.