TESIS

SENTIMEN ANALISIS BERBAHASA INDONESIA MENGGUNAKAN DEEP LEARNING

Disusun oleh:

Nama	**: Lilis Kurniasari**
NIM	**: 17.51.0982**
Konsentrasi	**: Business Intelligence**

PROGRAM STUDI S2 TEKNIK INFORMATIKA
PROGRAM PASCASARJANA UNIVERSITAS AMIKOM YOGYAKARTA
YOGYAKARTA

2020

TESIS

SENTIMEN ANALISIS BERBAHASA INDONESIA MENGGUNAKAN DEEP LEARNING

SENTIMENT ANALYSIS IN BAHASA USING DEEP LEARNING

Diajukan melalui Jalur Jurnal Bereputasi
untuk memenuhi salah satu syarat memperoleh derajat Magister

Disusun oleh:

Nama	**: Lilis Kurniasari**
NIM	**: 17.51.0982**
Konsentrasi	**: Business Intelligence**

PROGRAM STUDI S2 TEKNIK INFORMATIKA

PROGRAM PASCASARJANA UNIVERSITAS AMIKOM YOGYAKARTA

YOGYAKARTA

2020

HALAMAN PENGESAHAN

SENTIMEN ANALISIS BERBAHASA INDONESIA MENGGUNAKAN DEEP LEARNING

SENTIMENT ANALYSIS IN BAHASA USING DEEP LEARNING

Dipersiapkan dan Disusun oleh

Lilis Kurniasari

17.51.0982

Telah Diujikan dan Dipertahankan dalam Sidang Ujian Tesis
Program Studi S2 Teknik Informatika
Program Pascasarjana Universitas AMIKOM Yogyakarta
pada hari Rabu, 8 Juli 2020

Tesis ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Magister Komputer

Yogyakarta, 8 Juli 2020
Rektor

Prof. Dr. M. Suyanto, M.M.
NIK. 190302001

HALAMAN PERSETUJUAN

SENTIMEN ANALISIS BERBAHASA INDONESIA MENGGUNAKAN DEEP LEARNING

SENTIMENT ANALYSIS IN BAHASA USING DEEP LEARNING

Dipersiapkan dan Disusun oleh

Lilis Kurniasari

17.51.0982

Telah Diujikan dan Dipertahankan dalam Sidang Ujian Tesis
Program Studi S2 Teknik Informatika
Program Pascasarjana Universitas AMIKOM Yogyakarta
pada hari Rabu, 8 Juli 2020

Pembimbing Utama	**Anggota Tim Penguji**
Dr. Arief Setyanto, S.Si., M.T. **NIK. 190302036**	**Prof. Dr. Ema Utami, S.Si., M.Kom.** **NIK. 190302037**
	Dr. Andi Sunyoto, M.Kom. **NIK. 190302052**
	Dr. Arief Setyanto, S.Si., M.T. **NIK. 190302036**

Tesis ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Magister Komputer

Yogyakarta, 8 Juli 2020
Direktur Program Pascasarjana

Dr. Kusrini, M.Kom.
NIK. 190302106

HALAMAN PERNYATAAN KEASLIAN TESIS

Yang bertandatangan di bawah ini,

Nama mahasiswa	**: Lilis Kurniasari**
NIM	**: 17.51.0982**
Konsentrasi	**: Business Intelligence**

Menyatakan bahwa Tesis dengan judul berikut:
Sentimen Analisis Berbahasa Indonesia Menggunakan Deep Learning

Dosen Pembimbing Utama	: Arief Setyanto, S.Si., M.T., Ph.D.
Dosen Pembimbing Pendamping	: -

1. Karya tulis ini adalah benar-benar ASLI dan BELUM PERNAH diajukan untuk mendapatkan gelar akademik, baik di Universitas AMIKOM Yogyakarta maupun di Perguruan Tinggi lainnya
2. Karya tulis ini merupakan gagasan, rumusan dan penelitian SAYA sendiri, tanpa bantuan pihak lain kecuali arahan dari Tim Dosen Pembimbing
3. Dalam karya tulis ini tidak terdapat karya atau pendapat orang lain, kecuali secara tertulis dengan jelas dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan disebutkan dalam Daftar Pustaka pada karya tulis ini
4. Perangkat lunak yang digunakan dalam penelitian ini sepenuhnya menjadi tanggung jawab SAYA, bukan tanggung jawab Universitas AMIKOM Yogyakarta
5. Pernyataan ini SAYA buat dengan sesungguhnya, apabila di kemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka SAYA bersedia menerima SANKSI AKADEMIK dengan pencabutan gelar yang sudah diperoleh, serta sanksi lainnya sesuai dengan norma yang berlaku di Perguruan Tinggi

Yogyakarta, 8 Juli 2020
Yang Menyatakan,

METERAI TEMPEL
6000

Lilis Kurniasari

KATA PENGANTAR

Alhamdulillah, segala puji syukur penulis panjatkan kehadirat Allah SWT, atas segala karunia dan ridho-NYA, sehingga tesis dengan judul "Sentiment Analisis Menggunakan Recurrent Neural Network" ini dapat diselesaikan.

Tesis ini disusun untuk memenuhi salah satu persyaratan memperoleh gelar Magister Teknik Infomatika pada program studi Magister Teknik Informatika Universitas AMIKOM Yogyakarta.

Oleh karena itu, pada kesempatan ini penulis menyampaikan rasa hormat dan menghaturkan terima kasih yang sebesar-besarnya, kepada:

1. Bapak Arief Setyanto, S.Si, MT, Ph.D. selaku dosen pembimbing utama.
2. Ibu Dr. Kusrini, M.Kom selaku direktur Program Studi Magister Teknik Informatika Universitas Amikom Yogyakarta.
3. Segenap dosen dan karyawan Program Studi Magister Teknik Informatika Universitas Amikom Yogyakarta.
4. Bapak Muhammad Helmi Zain Nuri, M.T. dan bapak Dr. Dwijoko Purbohadi, M.T dosen Program Studi Teknologi informasi UMY.
5. Bapak ibu yang senantiasa mendoakan kelancaran studi.
6. Keluarga bapak Suharto yang telah memberikan dukungan selama masa studi.
7. Temen teman mahasiswa Program Studi Magister Teknik Informatika Universitas Amikom Yogyakarta.

Dengan keterbatasan pengalaman, ilmu maupun pustaka yang ditinjau, penulis menyadari bahwa tesis ini masih banyak kekurangan dan pengembangan lanjut agar benar benar bermanfaat. Oleh sebab itu, penulis sangat mengharapkan kritik dan saran agar tesis ini lebih sempurna serta sebagai masukan bagi penulis untuk penelitian dan penulisan karya ilmiah di masa yang akan datang.

Akhir kata, penulis berharap tesis ini memberikan manfaat bagi kita semua terutama untuk pengembangan ilmu pengetahuan yang ramah lingkungan.

Yogyakarta, 8 Juli 2020

Penulis

BAB I

PENDAHULUAN

Pemrosesan bahasa alami (NLP) telah menjadi salah satu bidang penelitian paling aktif selama dua dekade. Popularitasnya menarik para peneliti untuk memeriksa data, penambangan web, dan teks, dan pengambilan informasi [1]. NLP telah berkembang dari komputer ke ilmu manajemen dan sosial, menguntungkan bidang terkait seperti pemasaran, keuangan, politik, komunikasi, dan sejarah secara tidak penting. Opini memainkan peran penting dalam hampir semua aktivitas manusia dan memengaruhi perilaku. Ini menggerakkan kepercayaan, persepsi realitas, pilihan, dan sampai batas tertentu, berdampak pada cara orang lain melihat dan mengevaluasi dunia. Individu dan organisasi membutuhkan pendapat ketiga dalam pengambilan keputusan [2]–[5]. Karena itu, organisasi perlu mendengarkan pendapat pengguna tentang produk dan layanan mereka. Selain itu, ada beberapa cara untuk menganalisis opini, termasuk analisis sentimen. Ini adalah sistem yang secara otomatis mengidentifikasi dan memproses informasi, seperti teknik pembelajaran [2], [6]. Studi ini menguji bagaimana deep learning dapat diterapkan dalam analisis sentimen dalam Bahasa. Ini bertujuan untuk membantu organisasi memahami pendapat dan persepsi pelanggan tentang produk.

Penelitian ini menggunakan deep learning untuk menganlisi sentimen dari pengguna website traveloka. Algoritma yang digunakan adalah Recurrent Neural Network. Recurrent Neural Network berfungsi untuk mengklasifikasi sentiment. Data set yang digunakan dalam penelitian ini data review pengguna hotel pada website Traveloka. Data review pengguna Traveloka diproses menjadi vector kata dan data sets. Kami menggunakan word2vec untuk mengubah data review yang berupa string menjadi vector kata. Framework Tensorflow dan Bahasa pemrograman Python digunakan untuk menjalankan algoritma Recurrent Neural Network.

Penelitian pertama bertujuan untuk mengukur keakuratan model klasifikasi analisis sentimen menggunakan pembelajaran mendalam dan jaringan saraf. Penelitian ini menggunakan algoritma Recurrent Neural Network (RNN) dan Word2vec. Penelitian dimulai dengan membuat model klasifikasi analisis sentimen. Kemudian, model itu diuji melalui eksperimen. Dalam penelitian ini, kami menggunakan dua klasifikasi (positif dan negatif). Hasil penelitian menunjukkan bahwa pendekatan model memiliki akurasi yang lebih baik dengan model pembelajaran mesin lainnya dengan hasil akurasi adalah 91,98%.

Penelitian kedua mengusulkan Recurrent Neural Network (RNN) -Long Short Memory Term (LSTM) dalam mengklasifikasikan polaritas sentimen dalam bahasa Indonesia. LSTM diimplementasikan untuk membantu mengatasi komponen pertumbuhan vektor gradien yang berpotensi ada dalam algoritma RNN selama pelatihan [7]. Pertumbuhan vektor gradien menyebabkan jaringan RNN mengalami kesulitan dalam mempelajari urutan bersarang [7]. Representasi kalimat cukup sulit dalam pemrosesan bahasa alami, termasuk analisis sentimen. LSTM memiliki kinerja yang unggul dalam pemodelan kalimat yang berhubungan dengan tantangan tersebut [8]. Saat ini, dataset sentimen berlabel manusia dalam Bahasa Indoensia jarang tersedia. Dataset twitter dan Instagram hanya terdiri dari kurang dari sepuluh ribu kalimat [9], [10]. Penelitian ini selanjutnya mengevaluasi RNN - LSTM untuk mengenali sentimen kalimat polarisasi dalam teks bahasa Indonesia. Penelitian ini juga mengembangkan dataset sentimen dalam Bahasa, yang memiliki 25.000 kalimat, baik positif maupun negatif. Studi ini mengusulkan model untuk analisis sentimen menggunakan RNN-LSTM dan dataset bahasa Indonesia, dengan total ulasan sebanyak 25.000 kalimat yang telah dilabeli oleh manusia. Hasil dari penerapan algoritma RNN-LSTM pada data set berbahasa Indonesia menggunakan word2vec memiliki akurasi 95,0%, mengungguli CNN + word2vec, Naïve Bayes, dan RNN Conv.

BAB II

PUBLIKASI PERTAMA

Introduction

Currently, internet users in the world continue to experience rapid growth of technology. This increase is one of the factors triggering the evolution of social media and e-commerce. This evolution has caused an increase in the amount of information that is quite large or better known as big data. Today, information is a very important source for determining decisions, such as information about product reviews and ratings. Product reviews and ratings are very useful information for consumers to make decisions. Therefore, current sentiment analysis is an interesting topic to study. Analysis sentiment is one area of research in the field of NLP (Natural Language Processing) that classifies user reviews into positive and negative reviews. Analysis sentiment is the study of user reviews of the products or entities they use, such as food, hotels, airlines, etc [1], [2], [3]. The user's opinion on a product or entity has a very high influence on one's decision making. [4] said that nearly 95 percent of customers saw previous user reviews before they decided to buy or use products.

There are a lot of researches in the current field of analytical sentiment. Many algorithms are used in research in the field of sentiment analysis, for example noble of algorithms using traditional machine learning such as Support Vector Machine, Naïve Bayes, and learning using in-depth learning models [5], [6]. Research using deep learning shows better results such as the classification of text, images, sound, and video. The working principle of deep learning using a neural network follows the architecture of neural networks in the human brain. One of the neural network architectures used in analytic sentiment is recurrent neural networks (RNN). RNN algorithm will associate each word in the input with a certain time step. Simply put, the RNN will map the order of inputs into a vector of fixed size [7]. Therefore, this study uses the RNN architecture to calculate word dependence in sentences on NLP. Beside RNN, in this study also used word2vector to create word vectors.

This study evaluates and applies machine learning models using recurrent neural network algorithms and word2vec. These models used a set of data to test them. The data set used was a data sets containing reviews in Indonesian from the Traveloka website. Existing models would be used to classify user reviews into two categories, positive and negative reviews. The model would measure the level of accuracy with a minimum threshold of 91.9 percent.

Literature Review

The impact of the current internet development is that it is easy to get very large amounts of data. They can use the data to be analyzed to produce useful information [8], [9]. Several studies on analytical sentiment use traditional machine learning such as Support Vector Machine, Naïve Bayes etc. [1], [10], [11] use several traditional machine learning algorithms such as SVM and Naive Bayes in their researches to analyze sentiment from IMDB movie reviews.

Most of the previous studies use data sets in English and few use data sets in Bahasa Indonesian. [12], [13], [14], [15] conduct sentiment analysis research using data sets in English, Chinese and Indian. [16] use the RNN algorithm in their research to analyze sentiments in Malayalan language. The accuracy produced from the study is 80% and can still be improved using deeper data sets. [17] analyzes the performance of RNN and LSTM in classifying analytical sentiments for movie reviews. The results of this study show an accuracy of 86.74%. From this study, they find that the model is susceptible to overfitting.

[18] conduct research using CNN and word2vec to analyze sentiment on social media. They test framework using public data sets in film review corpus and produce an accuracy of 45.4%. [19] in his writing proposes an approach to analyze sentiments on product reviews using deep learning and word2vec. The main idea in his research is the use of word2vec to learn word embedding and deep learning to train product sentiment classes.

Methodology

The first steps in this research, we are scraping and scrawling the reviews on the Traveloka website. They use scraping and crawling techniques for data collection [20]. They process the review data into word vectors and data sets. The data sets in this study is divided into two parts, first as a training data sets and second testing data sets s. The flow of our research can be seen in figure 1.

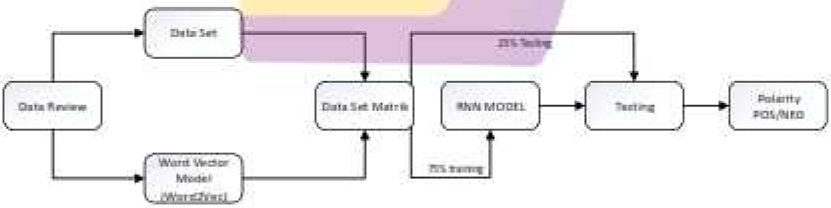

Figure 1. Framework of Model

a. Word Vector Model

From the process of scraping and scrawling we get structured data with string format. We must convert the data into vector form. We use word2vec to convert string words into vectors. Word2Vec is a tool developed by Google [21]. Word2vec will convert words into vectors by looking at the context of words with words that appear in the sentence. Other than that, word2vec will collect words with the same context that are close together in the vector space figure 2.

Figure 2. Synonym of good keyword

With context words in sentences They can find out whether these words are usually used in sentences that are positive or negative [22]. How word2vec works is by taking a set of sentence data in the corpus and by generating a vector for each word in the corpus. The output of word2vec is embedding matrix.

b. Recurrent Neural Network

This study uses the RNN algorithm to classify sentiments. RNN has a temporal aspect of the latter. This temporal aspect makes RNN different from feedforward neural networks. Figure 3 present that the RNN algorithm has the same number of time steps as the maximal sequence.

Figure 3. Number of Time Step

Beside the time step, there is a new component called the hidden state vector (ht). This vector has a function to encapsulate and to summarize all information in the previous time step. Whereas, vector xt has a function to summarize and to encapsulate information from certain words. The RNN summarizes the two vectors above into hidden status. Formula 1 shows the number of two vectors then entered into functions usually in the form of sigma or tan.

$$h_t = \sigma(w^H h_{t-1} + w^x x_t) \tag{1}$$

In the formula, w represents the weight matrix. Wh is a matrix that has the same weight in each time step, and Wx is a matrix that has different weights for each input. For each matrix weight, the magnitude affects the number of hidden state vectors and the current vector or previous state vector affects the hidden state vector. Then, the next process is to enter the last hidden state vector into the binary softmax classifier to produce negative or positive possibility of polarity. The above polarity is generated from the multiplication between hidden state vectors and the weight matrix, and then the results are entered into the softmax function

Results and Analysis

a. Experimental Word vector (Embedding matrix)

The word2vec model, has a matrix containing 136,281 vector words with dimensions of 300 for each vector. The model will be divided into two parts. The first part is a list of words used in python and the second part is an embedding matrix with dimensions of 136,281x300 which is used to hold all the values of the word vectors. This model collects words that have contexts that are almost as close as together in a vector space table 1.

Table 1. Word Vector

Positive	**Word**	*makan*	*makanan*	Sarapan	*breakfast*
	Vector	0.733680	0.733680	0.631776	0.631776
Negative	**Word**	*Jorok*	*Kusam*	*Kotor*	*dekil*
	Vector	0.527863	0.527863	0.385771	0.381265

The next step is to make a representation of each word vector. They use a Tensorflow function to take two arguments. The first argument is used to take the matrix from word2vec and the second argument to retrieve the index of each word. Vector index is an index of rows for each word such as in table 2.

Table 2. Index Word

sentence	*Saya suka kamar luas dekat kolam renang bersih dan segar*									
word	*saya*	*suka*	*Kamar*	*Luas*	*Deket*	*Kolam*	*Renang*	*Bersih*	*Dan*	*segar*
index	56	260	72	291	138	48	49	88	3	1253

b. Experimental Data Set

They use 5000 data sets to train the model. The data sets is divided into two, i.e. positive and negative. Each data sets is stored in a file with csv extension like in table 3.

Table 1. Review of Data set

	Positive review	**Negative review**
Total File	2,500	2,500
Total words	989,896	1,053,227
Average words	28,99	33,53

Figure 4 is a visual form with histogram format from a data set review. From the histogram, the average number of words is below 50. Then the max sequence value for our model is 50. The average number of sentences in the model is used as the max sequence length.

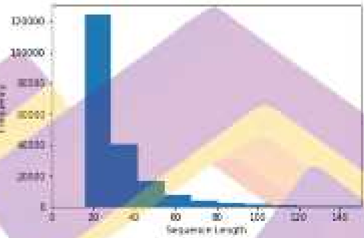

Figure 4. Histogram review data set

They create a set of matrices from the data sets above. RNN requires a collection of these matrices as input. From the training process They get 5000 x 50 matrices. Table 4 shows the example of index vector for a sentence. The data sets sets are then broken down by 75% for the training data sets and 25% for the testing data sets.

Table 4. Data set matrix

Sentence									
Staf sangat bagus, hotel nyaman seperti kita mengambil 3 kamar suite kerajaan dan rungan besar dan memiliki ruang terpisah. Hotel ini bagus, sarapannya enak, Cuma jalanya kecil, cukup untuk satu mobil. Semua baik baik saja									
Index vector									
403	44	185	37	828	146	154	1186	72	395
15900	3	478	329	3	802	491	37	39	185
212	33	45	2477	169	87	85	974	912	32
108	108	482	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

c. Experiment RNN

They used some hyperparameters for creating RNN model in table 5. Hyperparameter is used to improve the performance of the model.

Table 5. Hyperparameters

No	Hyperparameter	Total
1	Batch size	24
2	Number of clasess	2
3	Number of training iteration	10000

In this model, they use two placeholders. The first part is used for input into the RNN network while the second part is used for labels. The second part, placeholder labels represent examples of positive and negative training sets.

d. Result

In the training process, first They accommodate a collection of related reviews and labels. By using the run function, the network can define values where They can optimize components and minimize loss. in addition, the run function is used to feed a collection of reviews and labels. This process will be repeated as many as iterations have been set.

Figure 5. Accuracy and Loss

Figure 5 shows that pretrained model has accuracy approaching 92% percent and loss approaching 0 percent. As a comparison material in this study they also conducted a test using other algorithms such as CNN, Naive Bayes, RNN conv. Accuracy results can be seen in table 6. These results show that RNN with Word2vec can lead to better accuracy rate than the other model.

Table 6. Accuracy

Model	accuracy
CNN + Word2vec	0.8923
Naive Bayes	0.441
RNN Conv	0.8877
RNN + Word2vec	0.9198

Conclusion

In this research they have proposed model for sentiment analysis using RNN and word2vec. RNN in this model is implemented using framework Tensorflow. The research results show that the model approach has better accuracy with other machine learning models with result of accuracy is 91.98%. From this research, they also have to pay attention to the possibility of overfitting the model when carrying out the testing process. In the future They can try to use RNN and LSTM to overcome overfitting problems and to improve the performance of the model.

Reference

[1] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up? Sentiment Classification Using Machine Learning Techniques," in *Proceedings of the ACL-02 conference on Empirical methods in natural language processing - EMNLP '02*, 2002, vol. 10, pp. 79–86.

[2] A. Agarwal, B. Xie, I. Vovsha, O. Rambow, and R. J. Passonneau, "Sentiment Analysis of Twitter Data," in *Proceedings of the Workshop on Language in Social Media (LSM 2011)*, 2011, pp. 30–38.

[3] F. Poecze, C. Ebster, and C. Strauss, "Social media metrics and sentiment analysis to evaluate the effectiveness of social media posts," *Procedia Comput. Sci.*, vol. 130, pp. 660–666, 2018.

[4] C. Day, "The Importance of Sentiment Analysis in Social Media Analysis," *LinkedIn*, 2015.[Online].Available:https://www.linkedin.com/pulse/importance-sentiment-analysis-social-media-christine-day/. [Accessed: 01-May-2019].

[5] J. Bhaskar, K. Sruthi, and P. Nedungadi, "Hybrid Approach for Emotion Classification of Audio Conversation Based on Text and Speech Mining," *Procedia Comput. Sci.*, vol. 46, pp. 635–643, Jan. 2015.

[6] P. Singhal and P. Bhattacharyya, "Sentiment Analysis and Deep Learning : A Survey," *CoRR*, 2016.

[7] K. Cho, D. Bahdanau, F. Bougares, H. Schtheynk, and Y. Bengio, "Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1724–1734.

[8] B. Liu, Y. Dai, X. Li, W. S. Lee, and P. S. Yu, "Building text classifiers using positive and unlabeled examples," in *Third IEEE International Conference on Data Mining*, pp. 179–186.

[9] G. Li, S. C. H. Hoi, K. Chang, and R. Jain, "Micro-blogging Sentiment Detection by Collaborative Online Learning," in *2010 IEEE International Conference on Data Mining*, 2010, pp. 893–898.

[10] B. Pang and L. Lee, "Opinion Mining and Sentiment Analysis," *Found.*

Trends® Inf. Retr., vol. 2, no. 1–2, pp. 1–135, 2008.

[11] W. Feng, J. Sun, L. Zhang, C. Cao, and Q. Yang, "A support vector machine based naive Bayes algorithm for spam filtering," *2016 IEEE 35th Int. Perform. Comput. Commun. Conf. IPCCC 2016*, pp. 1–8, 2017.

[12] H. Shirani-Mehr, "Applications of Deep Learning to Sentiment Analysis of Movie Reviews," 2015.

[13] A. Timmaraju and V. Khanna, "Sentiment Analysis on Movie Reviews using Recursive and Recurrent Neural Network Architectures," *Semant. Sch.*, pp. 2–7, 2015.

[14] L. Zhang and C. Chen, "Sentiment Classification with Convolutional Neural Networks: An Experimental Study on a Large-Scale Chinese Conversation Corpus," in *2016 12th International Conference on Computational Intelligence and Security (CIS)*, 2016, pp. 165–169.

[15] P. Mishra, R. Rajnish, and P. Kumar, "Sentiment analysis of Twitter data: Case study on digital India," in *2016 International Conference on Information Technology (InCITe) - The Next Generation IT Summit on the Theme - Internet of Things: Connect your Worlds*, 2016, pp. 148–153.

[16] M. Thomas and L. C.A, "Sentimental analysis using recurrent neural network," *Int. J. Eng. Technol.*, vol. 7, no. 2.27, p. 88, 2018.

[17] F. Miedema, "Sentiment Analysis with Long Short-Term Memory networks," 2018.

[18] X. Ouyang, P. Zhou, C. H. Li, and L. Liu, "Sentiment Analysis Using Convolutional Neural Network," in *2015 IEEE International Conference on*

Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing, 2015, pp. 2359–2364.

[19] J. Panthati, J. Bhaskar, and T. K. Ranga, "Sentiment Analysis on Customer Reviews using Deep Learning," *Int. J. Comput. Sci. Eng.*, vol. 6, no. 7, pp. 1023–1024, 2018.

[20] P. Milev, "Conceptual Approach for Development of Theyb Scraping Application for Tracking Information," *Econ. Altern.*, no. 3, pp. 475–485, 2017.

[21] X. Rong, "word2vec Parameter Learning Explained," Nov. 2014.

[22] A. Deshpande,"Perform sentiment analysis with LSTMs, using TensorFlow - O'Reilly Media." [Online]. Available: https://www.oreilly.com/learning/perform-sentiment-analysis-with-lstms-using-tensorflow. [Accessed: 20-Apr-2019].

BAB III

PUBLIKASI KEDUA

Introduction

Natural language processing (NLP) has been one of the most active research areas for two decades. The popularity of NLP attracts many researchers to learns about data mining, web mining, text mining, and information retrieval [1]. It has even expanded from computer science to management and social sciences. Many related fields such as marketing, finance, politics, communication and even history start to gain benefits from natural language processing. Opinion plays an important role in almost all human activities and affects our behavior. Opinion drive our beliefs, perceptions of reality, the choices we make, and to a certain extent, are conditioned on the way others see and evaluate the world. Some individuals and even organizations need the opinions of others to be taken into consideration in deciding what decisions or steps to take [2–5]. Therefore, organizations need to listen to users' opinions about the products and services they provide. There are several ways to analyse user opinions, one of which is sentiment analysis. Sentiment analysis systems can automatically identify and process information, such as learning techniques [2,6]. We will examine how deep learning implemented in sentiment analysis in Bahasa. This study is done to help organizations get a picture of their product.

One area in machine learning is structured learning which is commonly known as deep learning and hierarchical learning. For each problem and learning technology there are various deep learning architectures that can be used [7,8]. Each architecture is built on a network that is different from the function of each. One of the architectures is recurrent neural network (RNN). NLP has a temporal aspect where the word in the sentence has a dependence on the word before and after it, this research used RNN to reduce the dependence of each word in a sentence. RNN associates each word in the order of input sentences with a certain time step. Each number of time steps will be equal to the maximum sequence length of each word. Along with the time step there is a new component called hidden vector. Hidden vector summarizes all information in the previous time steps. Traditionally, a simple strategy for sequencing modelling is to map the order of inputs to fixed-size vectors using one RNN [9]. We also implemented Long Short-Term module (LSTM). This module is useful for overcoming the problem of growth of vector gradient components that potentially exist in the RNN algorithm during the training period [10]. Gradient vector growth has caused the RNN network to experience difficulties in studying sequences in sequence [11]. In analytical sentiments there are challenges faced when using deep learning that is modelling representations of words or sentences, LSTM has a superior performance in handling these challenges [12]. The purpose of this research is to evaluate and apply deep learning models, specifically, RNN-LSTM. This is achieved using the user reviews dataset from the Traveloka website, which receives 28.92 million visitors and 78.49% user traffic in Indonesia [12,13]. In 2019, it had more than 500,000 hotel user reviews in Bahasa. This research considers positive and negative opinion polarization in training and testing the model. Most of the existing work on natural language processing (NLP), including sentiment analysis, uses English text [14,15]. RNN - LSTM had excellent performance in several studies on sentiment analysis using the English text dataset [16,17]. Currently, the human labelled sentiment dataset in Bahasa rarely available. Dataset twitter and Instagram consist of only less than ten thousand sentences [18,19]. This study furthermore evaluates the RNN - LSTM to recognize a sentence sentiment polarization in Indonesian text. It also develops a sentiment

Literature Review

There are several previous studies regarding sentiment analysis using algorithms such as Naive Bayes. Bingwei [18] examines the scalability of Naive Bayes classifier on big data to achieve fine-grain control from the analysis procedure in the movie review dataset. Alec Go [19] used three algorithms namely Naive Bayes, maximum entropy, and support vector machine to classify the sentiment of twitter messages using distant supervision. Some research where scientists use deep learning and neural networks to analyse sentiments but most of these studies uses in English [15,20], and Asian languages [21–24].

Merin Thomas [16] apply a recurrent neural network to analyse sentiment from tweeters in southern Malayalam language. The results showed that the accuracy of the model made using the RNN-LSTM method was 80%. This accuracy value can still be improved by using a deeper dataset. What's interesting about this research is that this method can be implemented using other languages. The RNN-LSTM method is two models that are often used to analyse sentiment.

Fenna Miedema [17] in a study entitled Sentiment Analysis with Long Short-Term Memory Networks conducted a study to find out why the RNN-LSTM model works well for analysing sentiments and how these models perform. LSTM is used to classify sentiments with a movie review dataset. The results show that the model can correctly classify 86.74% of the total reviews into the validation set. This model is very sensitive to overfitting but provides good results even without setting parameters.

Xin Wang and Yuanchao Liu [25] state that traditional RNN is not strong enough to handle complex sentiment expressions. Therefore, LSTM is implemented to classify sentiments. Experiment was done with a corpus of tweeters, a dataset containing 800,000 tweets labelled positive and negative. The results show that the LSTM network outperforms all other methods including SVM and Naïve Bayes.

A few research have been conducted on sentiments analysis in Indonesian text using machine learning techniques. Watrianthos [12] used a dataset from the Traveloka user review on the play store. This study aims to determine user perceptions based on service quality measurements using the Naïve Bayes method. Iswanto [27] used Naïve Bayes classifiers, Maximum Entropy classifiers, and Support Vector Machines in preprocessing the twitter dataset. This method achieves the recall and precision of up to 85.50%. Kurniawan [28] research using a 1.720 dataset in Bahasa had an average F measure of 75.18%. Hemanto [18] used a Naïve Bayes method for classifying twitter user reviews in Bahasa. This research classifled reviews in positive and negative aspects. The details are summarized in Table 1.

Table 1. Literature review resume

Reference	Dataset positive/Negative/ neutral	Language	Classifier	Acc (%)
Shirani-Mehr [21]		English	CNN + word2ve	46%
Bingwei et.al [20]	1000/1000	English	Naive Bayes	82%
Alec Go et al. [14]	800k/800k	English	SVM	81%
Timmaraju Aditya [15]	5000/5000	English	SMV Linier	86.49%
			2-layer NN	83.94%
			RecNN-RNN	83.88%
Al-smadi et.al [22]	24028	Arabic	SVM	95.40%
			K-Nearest Neighbor	94.10%
Lei Zhang [23]	929/946	Chinese	CNNs	88.75%
M. Heikal [24]	2500/2500	Arabic	CNN + LSTM	64.46%
Pasupa [25]	309/298/508	Thai	CNN	71.40%
			LSTM	60.20%
			Bi-LSTM	60.50%
Merin thomas [16]	2500/2500	Malayalam	RNN + LSTM	80%
Fimma Miedema [17]	50000/50000	English	RNN + LSTM	86.74%
Watmanthos [12]		Bahasa	Naive Bayes	31.02%
Irwanto [27]		Bahasa	Support Vector Machines	85.50%
Kurniawan [28]	430/430/860	Bahasa	Naive Bayes	75.18%
Hermanto [18]		Bahasa	Naive Bayes	

Methodology

Figure 1 illustrates the proposed framework. A total of 1,8 million reviews were collected from the traveloka website during 2018 august. The data is used as a corpus to create word vectors. Positive and negative 25000 reviews and labels were randomly selected. The labeled reviews are used as training and testing data sets. Word2vec is a two-layer neural network that processes words in the corpus into a collection of vectors for the inner network to understand them. Our model uses an RNN-LSTM algorithm to classify user reviews. Word2vec transformation needs to transform the word into a vector to satisfy the required input of the LSTM network.

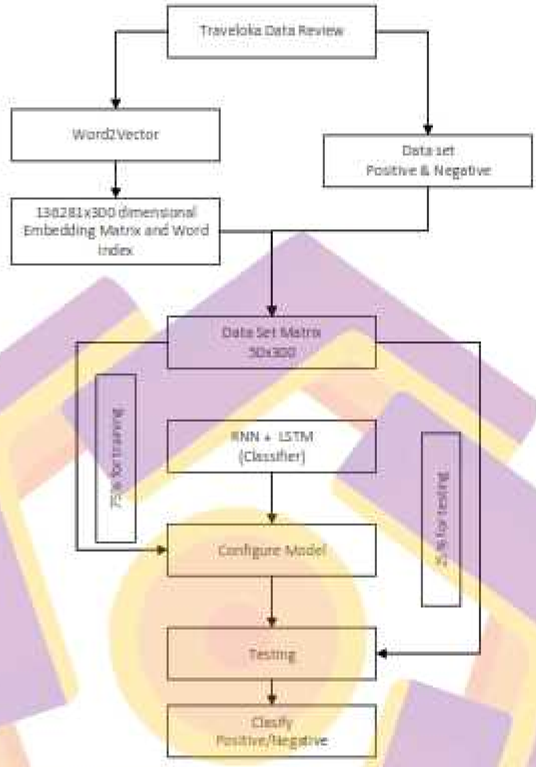

Figure 1. The Proposed Framework

Apart from creating word vectors, word2vec group vectors from words with definitions and uses in the same context into vector space. Word2vec detects every word vector mathematically. Words with the same distance are often close to each other in vector space. Representation of word vectors is called embedding words. Fairly accurate guesses can be made on the meaning of the word with enough data and context. This improves the performance of the recurrent neural network algorithm used in the model.

a. Word2Vec

Google developed word2vec initially implemented two algorithms called skip-gram models and a continuous bag of the word (CBOW) [29]. Word2vec takes a word from the corpus as input and produce word vector as output. It works optimally in guessing words accurately if there are sufficient data, usefulness, and context of each word.

We trained the word2vec model using 1,8 million review data from the Traveloka website with 138280 words. Words often used in the same context are in a similar group in vector space. Grouping words by context help the model to reduce error rates. The model produces a word vector, each representing the word and its context. Besides, it also represents the word meaning and semantics [30].

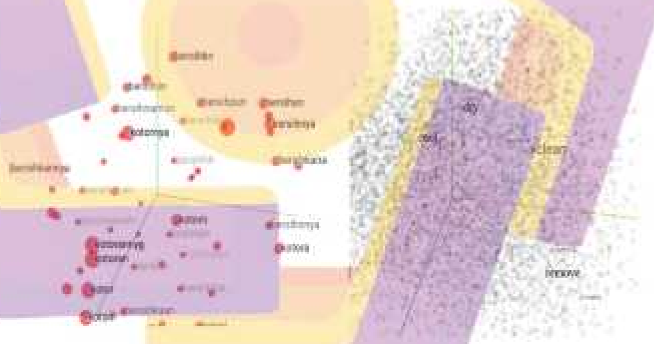

Figure 2. Word with same context (left in Bahasa, right in English)

Figure 2 shows an example of the implementation of the word2vec model in Bahasa and English. On the left, a piece of Word2vec result in Bahasa, it group Bersih (clean) and Kotor (dirty) words. Semantically, closer words are located firmly in the vector space on the right side of a word2vec for English. Table 2 shows the word vector in Bahasa and English. Bersih is adjacent to the word vector Bersih, Bersih, Bersih, bersi, and clean close to bright, shining, fresh, while dirty close to dusty, mesy, and muddy. RNN requires this cluster for the starting point of training.

Table 2. Words Vector

Bahasa	**Word**	*Bersih*	*bersihkan*	*Kotor*	*kotoran*
	Vector	0.380360	0.379824	0.527863	0.460976
English	**Word**	Clean	cleaning	dirty	dirt
	Vector	0.658972	0.678098	0.689567	0.690235

Word vector model produces a matrix with 136,281 words vectors, each with a dimension of 300. Two data structures were created for the training process. Firstly, 136,281 words vectors were made to a list in Python. Secondly, 136,281 x 300 embedding matrices to load all values of the word vectors were made.

The next step involves making a vector representation of the input sentence. The function of Tensorflow embedding was utilized to get word vectors. Embedding tensor has a function for retrieving the embedding matrix. It is also useful for retrieving the index of each word in the sentence. The index of each word also represents the index of the word vector. This is basically a row index of each word in the input sentence. Table 3 shows an example index word vector for a sentence "pagi pagi sarapan nasi goreng di Kasur sambil nonton renang".

Table 3. Index Words Vector

sentence	*Pagi Pagi sarapan nasi goreng di Kasur sambil nonton renang*								
word	*Pagi*	*Pagi*	*Sarapan*	*Nasi*	*Goreng*	*Di*	*Kasur*	*Sambil*	*Nonton*
index	548	548	130	225	226	19	461	1068	2669

b. Recurrent Neural Network

Recurrent neural network architecture overcomes temporal aspects in NLP. This is where the word in the sentence depends on the word before or after it. In the RNN structure, each word in the sentence is connected to a certain time in sequence according to the input. Therefore, the number of steps is the same maximum length of the word sequence in the sentence, as shown in table 4.

Table 4. Number of Time Step

sentence	Pagi	Pagi	Sarapan	...	Nonton
Sequence length	X_0	X_1	X_3		X_9
Time	t=0	t=2	t=3		t=4

RNN applies current and past input sources for the process. Figure 4 explains the sentence "pagi pagi sarapan nasi goreng di Kamar" as an input at the moment and CONTEXT UNIT as the output of the previous moment or input in the past. The last moment output in time step or t-1 has an impact on the current moment's output at the time step or t.

$$h_t = \sigma(w^H\, h_{t-1} + w^x\, x_t) \tag{1}$$

However, the RNN maintains this sequential information in a hidden state. This process achieves many time steps to impact the new process. It is formulated into a mathematical formula (1), where ht hidden state is a function of the current input, multiplied by the weight matrix w, then added input to the past ht-1 multiplied by the transition matrix u. Weight matrices determine adjustments to current input and the previously hidden conditions. Errors that occur are reprocessed by backpropagation until the lowest error rate is reached.

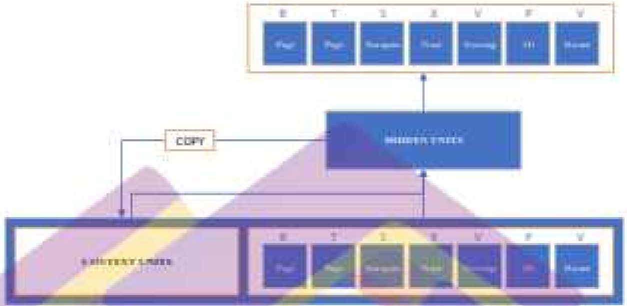

Figure 3. Input RNN

To update the weight matrix, Adam optimization is used to process backpropagation through time. Afterward, the hidden state vector in the last time step is entered into the Binary Softmax classifier, which produces values of 0 and 1, or provides the possibility of positive or negative sentiment as shown in figure 4. Tensorflow with CUDA was used in the training model. The device supporting the training process uses two GPUs from Invidia 1070 ti. KERAS is used in the deep learning process.

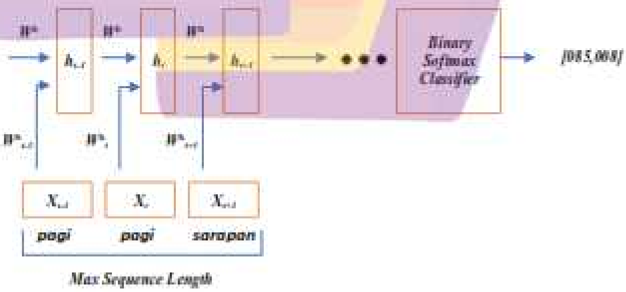

Figure 4. Binary Softmax Classifier

c. Long Short-Term Memory Units (LSTM)

Long Short-Term Memory is a module in an RNN network that solves missing gradient problems. In general, RNN applies the LSTM network to avoid propagation errors. This enables the RNN learning through many steps of time. LSTM contains cells that store information outside a recurring network. The cell is like the memory in the computer, deciding when the data needs to be stored, written, read, or deleted through the gate, as shown in figure 5. There are four gates that LSTM use, including an input, forget, and output gates, and a new memory container [30].

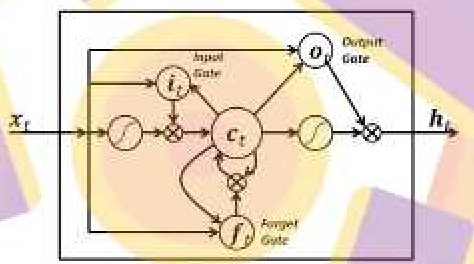

Figure 5. LSTM Gate

The study uses four hyperparameters to improve the performance of RNN and LSTM effectively. The learning rate starts from 0.001, and it is used by RNN to maintain fluctuations and training processes. Adam optimizer is utilized since it has properties with adaptive learning levels. Also, the 64 LSTM units are used for this model training, though the number of units depends on the average length of each review. Finally, from the previous embedding matrix process, the size of the word vector 136281 x 300 is used.

Results and Discussion

The prediction of the aerodynamic coefficients of the investigated projectiles shown in Fig. 1 was carried using the methods and the computer programme described above. The effects of forebody and afterbody shapes on the aerodynamics at supersonic speeds are analysed in this paper.

a. Data Set

This study used the Traveloka travel web and mobile application review dataset. It consisted of twenty-five thousand datasets divided into positive and negative classes, each with 12,500. The dataset is stored in a CSV file. Table 5 shows the review dataset.

Table 5. Review Data Set

	Positive review	**Negative review**
Total reviews	12.500	12.500
Total words	2.989.896	3.353.227
Average words	28.99	33.53

Figure 7 shows a histogram of the sentence length distribution of the review dataset. Matplot library was used to visualize the data. We apply the visualization histogram to know the average number of words each review. We use the average word value as the value of the max sequence length of the model. Figure 6 shows the distribution of sentence length distributed between 20 to 140 words. According to the histogram, the average review is under 50 words. The average value is used as the max sequence length value at 50. Therefore, an input of over 50 words is truncated.

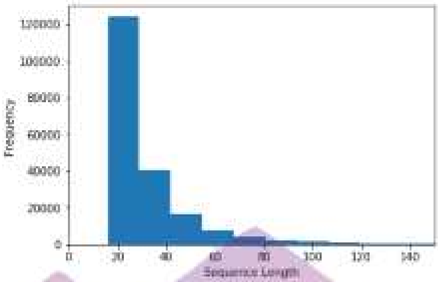

Figure 6. Histogram Review of Data Set

b. Experimental Data Set Matrix

The model used four hyperparameters to improve the performance, as shown in table 6. In this work, there is a need to specify placeholders. Two placeholders were created, one as input into the network and the other as labels.

Table 6. Hyperparameters

No	Hyperparameter	Total
1	Batch size	24
2	LSTM unit	64
3	Number of classes	2
4	Number of training iteration	10000

Labels' placeholders contain a set of [1.0] values representing a positive review dataset and [0.1] representing a negative review dataset. Placeholders have rows that represent the input of each training dataset. The model also uses placeholders to hold input data as input into the system, as shown in figure 7.

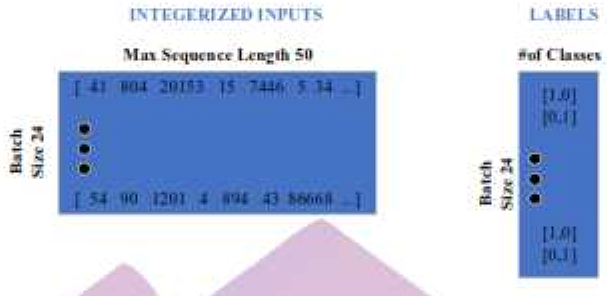

Figure 7. Placeholder on LSTM

With placeholders, the model runs the TensorFlow lookup function to capture word vectors generated by word2vec in the previous process. The word vectors, also called embedding matrices, with placeholders return the 3-D batch tensor dimensions with the maximum dimension sequence length of the embedding matrix shown in figure 8. Using the 3-D test makes it easier to visualize the input data in an integer form on TensorFlow.

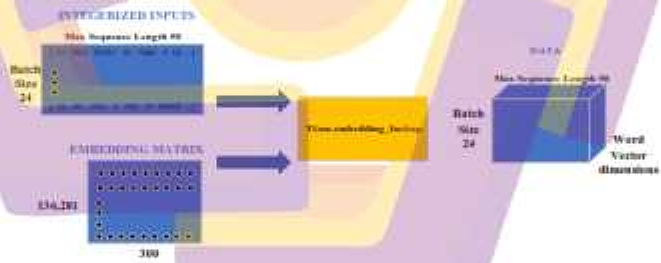

Figure 8. 3-D Dimension

The results of the above process form data as integers and input into the LSTM network. The integer data is placed into the LSTM unit used with a function in Jupiter. After the LSTM is filled with data, the system wraps the cell in the dropout layer to prevent overfitting. The selection of LSTM network architecture helps to utilize hidden state vectors. This process is achieved by stacking a lot of LSTM cells. This ensures the model maintains information from long-term dependence; otherwise, it can be used to provide parameters to the model. Therefore, the LSTM network usually takes a little longer in the learning process and more training examples. Figure 9 shows an overview of the model. The input is an embedding matrix with a max sequence length of 50. Afterward, 64 LSTM cell with four gates each was used. The output layer using binary softmax process 64 LSTM outputs to make predictions [1.0] or [0.1].

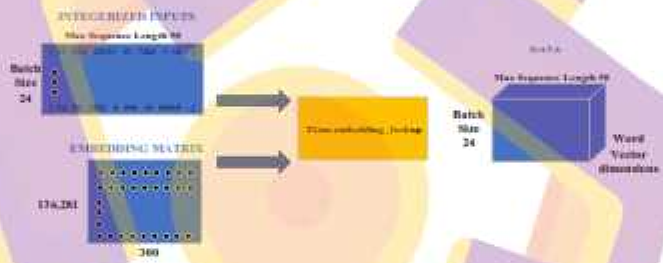

Figure 9. LSTM Network

The first output of the dynamic RNN function can be considered as the last hidden state vector. This vector might be reshaped and multiplied by a final weight matrix and a bias term to obtain the final output values [25]. The model ensures the prediction formulation works correctly using the maximum value index of all two output values matched with the training data set label.

c. Result and analysis

The first analysis step involves defining Tensorflow and loading the reviews and labels, then running function in the training session. This function has two arguments, specifically fetches and feed_dict. Essentially, the fetches argument defines the value of the computational process, while the feed_dict argument enters the data. Both arguments are for floating all input from the placeholders. The data structure serves as input to batch reviews and labels. Also, the training process repeats according to the number of iterations, which are set to 10000 iterations. From the training set, an accuracy of 0.950% and the loss value of close to zero% was obtained.

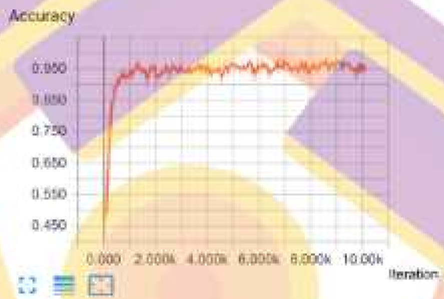

Figure 10. Model Training Accuracy

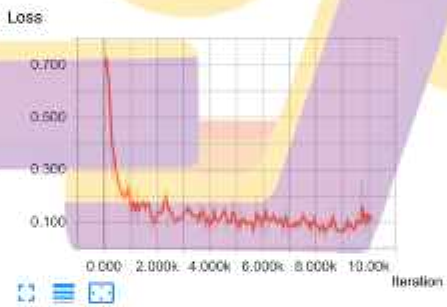

Figure 11. Model Training Loss

From Figure 11 and 10, the losses decrease while performance accuracy increases. However, it should be monitored for possible overfitting during training, which is a common phenomenon in machine learning [27]. It occurs when the model becomes compatible with the dataset to reduce losses to zero. Therefore, training stops before the loss value reaches zero to anticipate overfitting based on intuition techniques. For comparison, training using other machine learning algorithms such as CNN, Naive Bayes, and conventional RNN was conducted. The evaluation result of each algorithm is shown in table 7. These results indicate that RNN-LSTM achieves better accuracy compared to the other model. Thew accuracy cannot be directly compared since the dataset is different.

Table 7. Result Comparison

Model	accuracy	dataset
CNN + Word2Vec [31]	0.91323	UCI & Kaggle
Naïve Bayes [31]	0.441	UCI & Kaggle
RNN Conv [32]	0.8977	WWW.JD.COM
RNN + LSTM	0.950	www.traveloka.com

A model validation test was conducted using four-fold cross-validations. The dataset was divided into four equal sizes, as shown in table 8. Furthermore, four experiments were conducted with composition, as shown in table 9.

Table 8. Data Set Group

Fold 1	Fold 2	Fold 3	Fold 4
25%	25%	25%	25%

Table 9. Experiment Cross-Validation

Experiment	Training			Testing
exp 1	Fold 1	Fold 2	Fold 3	Fold 4
exp 2	Fold 1	Fold 2	Fold 4	Fold 3
exp 3	Fold 1	Fold 3	Fold 4	Fold 2
exp 4	Fold 2	Fold 3	Fold 4	Fold 1

The validation test results are shown in Table 10. From the validation test, the average value of accuracy of 94.88% and a standard deviation of 0.4426 were obtained.

Table 10. Confusion matrix

Exp	Accuracy	Precision	Recall	F1 Score
1	94.44%	90.21%	97.81%	93.64%
2	96.11%	96.78%	93.97%	95.73%
3	93.74%	94.45%	89.98%	92.45%
4	95.24%	92.87%	96.51%	94.89%

Conclusions

This study proposed a model for sentiment analysis using RNN-LSTM for the Indonesian language dataset with 25,000 human labelled sentences. The RNN in the model is implemented using the TensorFlow framework which helps to classify sentiments and understand the natural languages. Moreover, the results from the experiments showed that RNN-LSTM has 95.0% accuracy, outperforming CNN+word2vec, naïve Bayes, and RNN Conv. In the training process, intuition techniques were used to solve overfitting problem that often occurs during the training process. However, scientists have recently proposed grid LSTM with high performance in some applications, though it should be used to improve the performance of the sentiment classifier in future. This research used only positive and negative data sets. Future projects should use the three label data sets with mine neutral to evaluate the performance of the model.

Nomenclatures

ht	Hidden State of Time Step
W^{st}	Matrix-based on Previous Hidden State
W^{x}	Matrix-based on the Current Input
X_t	Time Step
h_{t-1}	Hidden State of the Previous Time Step
i_t	Input Gate
O_t	Output Gate
F_t	Forget Gate
C	New Memory Container
Greek Symbols	
σ	Tanh.
Abbreviations	
NLP	Natural Language Processing
RNN	Recurrent Neural Network
LSTM	Long-Short Term Memory
SVM	Support Vector Machine
CNN	Convolution Neural Network
CSV	Comma-Separated Values

References

1. Rezaeinia, S.M.; Rahmani, R.; Ghodsi, A.; and Veisi, H. (2019). Sentiment analysis based on improved pre-trained word embeddings. *Expert Systems with Applications*, 117, 139–147.

2. Zhang, L.; Wang, S.; and Liu, B. (2018). Deep Learning for Sentiment Analysis : {A} Survey. *CoRR*, abs/1801.0, 32.

3. Appel, O.; Chiclana, F.; Carter, J.; and Fujita, H. (2017). A Consensus Approach to the Sentiment Analysis Problem Driven by Support-Based IOWA Majority. *International Journal of Intelligent Systems*, 32, 947–965.

4. Appel, O.; Chiclana, F.; Carter, J.; and Fujita, H. (2017). Cross-ratio uninorms as an effective aggregation mechanism in sentiment analysis. *Knowledge-Based Systems*, 124, 16–22.

5. Appel, O.; Chiclana, F.; Carter, J.; and Fujita, H. (2018). Successes and challenges in developing a hybrid approach to sentiment analysis. *Applied Intelligence*, 48, 1176–1188.

6. Cambria, E. (2016). Affective Computing and Sentiment Analysis. *IEEE Intelligent Systems*, 31, 102–107.

7. Song, M.; Park, H.; and Shin, K. shik. (2019). Attention-based long short-term memory network using sentiment lexicon embedding for aspect-level sentiment analysis in Korean. *Information Processing and Management*, 56, 637–653.

8. Dosoula, N.; Griep, R.; and Den Ridder, R. (2016). Sentiment analysis of multiple implicit features per sentence in consumer review data. *Frontiers in Artificial Intelligence and Applications*, 291, 241–254.

9. Cho, K.; Bahdanau, D.; Bougares, F.; Schwenk, H.; and Bengio, Y. (Association for Computational Linguistics, 2014). Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* 1724–1734.

10. Kolen, J.F.; and Kremer, S.C. (IEEE, 2009). Gradient Flow in Recurrent Nets: The Difficulty of Learning LongTerm Dependencies. in *A Field Guide to Dynamical Recurrent Networks*.

11. Hochreiter, S.; and Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, 9, 1735–1780.

12. Watrianthos, R.; Suryadi, S.; Irmayani, D.; Nasution, M.; and Simanjorang, E.F.S. (2019). Sentiment Analysis Of Traveloka App Using Naïve Bayes Classifier Method. *International Journal of Scientific & Technology Research*, 8, 786–788.

13. Website Performance (2019). worldwide-overview. Retrieved 15 November 2019, From https://pro.similarweb.com/#/website/worldwide-overview/traveloka.com/*/999/3m?webSource=Total.

14. Go, A.; Bhayani, R.; and Huang, L. (2009). Twitter Sentiment Classification using Distant Supervision. *CS224N Project Report*.

15. Timmaraju, A.; and Khanna, V. (2015). Sentiment Analysis on Movie Reviews using Recursive and Recurrent Neural Network Architectures. *Semantic Scholar*, 2–7.

16. Thomas, M.; and C.A, L. (2018). Sentimental analysis using recurrent neural network. *International Journal of Engineering & Technology*, 7, 88.

17. Miedema, F. (2018). *Sentiment Analysis with Long Short-Term Memory networks*. Research paper. Vrije Universiteit Amsterdam.

18. Hermanto, D.T.; Ziaurrahman, M.; Bianto, M.A.; and Setyanto, A. (2018). Twitter Social Media Sentiment Analysis in Tourist Destinations Using Algorithms Naive Bayes Classifier. in *Journal of Physics: Conference Series* 1140,

19. Rosanensi, M.; Madani, M.; Wanggono, R.T.P.; and Setyanto, A. (l2018). Analysis sentiment and tourist response to rinjani mountain tour based on

comments from photo upload in instagram. in *Proceedings - 2018 3rd International Conference on Information Technology, Information Systems and Electrical Engineering, ICITISEE 2018* 184–188.

20. Liu, B.; Blasch, E.; Chen, Y.; Shen, D.; and Chen, G. (2013). Scalable sentiment classification for Big Data analysis using Naïve Bayes Classifier. *Proceedings - 2013 IEEE International Conference on Big Data, Big Data 2013*, 99–104.

21. Shirani-Mehr, H. (2015). *Applications of Deep Learning to Sentiment Analysis of Movie Reviews*. research paper. Department of Management Science & Engineering Stanford University.

22. Al-smadi, M.; Al-ayyoub, M.; Jararweh, Y.; and Qawasmeh, O. (2019). Enhancing Aspect-Based Sentiment Analysis of Arabic Hotels ' reviews using morphological , syntactic and semantic features. *Information Processing and Management*, 56, 308–319.

23. Zhang, L.; and Chen, C. (IEEE, 2016). Sentiment Classification with Convolutional Neural Networks: An Experimental Study on a Large-Scale Chinese Conversation Corpus. in *2016 12th International Conference on Computational Intelligence and Security (CIS)*, 165–169.

24. Heikal, M.; Torki, M.; and El-Makky, N. (2018). Sentiment Analysis of Arabic Tweets using Deep Learning. *Procedia Computer Science*, 142, 114–122.

25. Pasupa, K.; and Seneewong Na Ayutthaya, T. (2019). Thai sentiment

analysis with deep learning techniques: A comparative study based on word embedding, POS-tag, and sentic features. *Sustainable Cities and Society*, 50, 101615.

26. Wang, X.; Liu, Y.; SUN, C.; Wang, B.; and Wang, X. (Association for Computational Linguistics, 2015). Predicting Polarities of Tweets by Composing Word Embeddings with Long Short-Term Memory. in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, (Volume 1: Long Papers)*, 1343–1353.

27. Iswanto, B.H.; and Poerwoto, V. (2018). Sentiment analysis on Bahasa Indonesia tweets using Unibigram models and machine learning techniques. in *IOP Conference Series: Materials Science and Engineering*, 434, 434.

28. Kurniawan, S.; Kusumaningrum, R.; and Timu, M.E. (2019). Hierarchical Sentence Sentiment Analysis of Hotel Reviews Using the Naïve Bayes Classifier. *2018 2nd International Conference on Informatics and Computational Sciences, ICICoS 2018*, 104–108.

29. Rong, X. (2014). word2vec Parameter Learning Explained. *ArXiv, 1411.*

30. Deshpande, A.(2018). Perform sentiment analysis with LSTMs, using TensorFlow - O'Reilly Media. Retrieved 20 April 2019, from https://www.oreilly.com/learning/perform-sentiment-analysis-with-lstms-using-tensorflow.

31. Panthati, J.; Bhaskar, J.; and Ranga, T.K. (2018). Sentiment Analysis on

Customer Reviews using Deep Learning. *International Journal of Computer Sciences and Engineering*, 6, 1023–1024.

32. Li, D;. and Qian, J. (2016). Text sentiment analysis based on long short-term memory. *2016 1st IEEE International Conference on Computer Communication and the Internet, ICCCI 2016*, 471–475.