Journal of Harbin Engineering University
ISSN: 1006-7043

Vol 45 No. 6
June 2024

The Judgement of the Reviews of the Movie Oppenheimer by the LSTM Model in Deep Learning

Namkil Kang
Professor
College of Liberal Arts
Far East University
South Korea

Abstract
The main purpose of this paper is to make the LSTM model predict whether 100 reviews of the movie Oppenheimer are positive or negative. We obtained 100 reviews of the movie Oppenheimer in the portal site Naver in 2024. While we classified 70 reviews into train data, we classified 30 reviews into test data. Also, we classified 20% of train data into validation data. We used a hidden layer to increase the accuracy of the LSTM model. We used sigmoid as activation and rmsprop as optmizer. A major point of this paper is that there was a gradual fall in the loss of train data. More specifically, there was a continual decline in the loss of train data from epoch 1 to epoch 12. However, there was a slight rise in the loss, but there was a steady decrease in the figure from epoch 14 to epoch 25. A point to note is that the accuracy rate of the LSTM model reached a peak when epoch was 7 (92.86%). A further point to note is that the val_loss of validation data increased to 0.8977 when epoch was 12. It is worthwhile pointing out that the val_accuracy of validation data did not improve even though learning took place 25 times. It increased to 64.64%, but there was a slight decline in the val_loss of validation data when epoch was 12.
Keywords: LSTM, review, movie, Oppenheimer, loss, accuracy, val_loss

1. Introduction

The main goal of this paper is to train the LSTM model to predict whether 100 reviews of the movie Oppenheimer are positive or negative. We obtained 100 reviews of the movie Oppenheimer in the portal site Naver in 2024. We classified 70 reviews into train data, whereas we classified 30 reviews into test data. Also, we classified 20% of train data into validation data. We used a hidden layer to increase the accuracy of the LSTM model. We made the LSTM model learn words and predict whether or not each review is positive. By using RNN, we made our model treat data sequentially. The so-called RNN recognizes the relationships among words, thereby making our model judge each review as positive or negative. We assigned each index to each word so that our model can recognize the relationships among words. First, we aim to provide 30 samples of test data and those of train data. As stop words, we eliminated 22 words. Second, we assigned each number to each word in descending order to make the LSTM model judge each review as positive or negative. Simply put, we assigned each number to 320 words. Third, we made the LSTM model judge each review in train data and validation data as positive or negative. Put differently, learning took place 25 times for the judgement of the LSTM model. Fourth, we aim at providing the classification report on test data in which we provide the so-called accuracy, precision, and recall. Fifth, we provide two graphs for the accuracy and val_accuracy and the loss and val_loss.

2. Results

2.1. Samples of test data and train data

In this section, we provide 30 samples of test data and those of train data for expository purposes. We obtained 30 test data (30 reviews) and 70 train data (70 reviews) in the portal site Naver. For expository purposes, we translated them into English:

Table 1 Samples of test data

Number	Review	Label
1	Long running time	0
2	A weak point is that it took long time	0
3	Boring and not fun	0
4	I didn't understand the movie for two hours	0
5	I was engrossed in the movie	1
6	I expected it, but long running time	0
7	A masterpiece. A real masterpiece	1
8	What a shame!! If I consider Nolan's previous movies	0
9	Even though the movie is Nolan's, it is too long	0
10	I enjoyed Nolan's direction and music	1
11	A good movie for my life	1
12	It seems to me that Nolan will win the Academy award for Best Picture	1
13	A kind of biography that lacks fun	0
14	The most breathless movie that I have ever seen	1
15	A good movie, but there is no popularity	0
16	Too long and boring	0
17	I can't stand it because it is too long	0
18	Dizzy and boring	0
19	The movie was interesting and excellent	1
20	I couldn't stand it because it took long time	0
21	Actors' acting was good enough, but it took long time	0
22	I enjoyed the last part of the movie very much for an hour	1
23	I had a hard time because of long running time	0
24	The best one that I have ever seen	1
25	Nolan's ability that makes me sit still for 3 hours	1
26	I couldn't figure it out	0
27	Boring because of long running time	0

28	I was touched because of the last words	1
29	Very boring, so I got out	0
30	I slept well. Very boring	0

It is worthwhile noting that as illustrated in Table 1, we assigned 1 or 0 to each review, depending on its meaning. Put differently, while a positive one is 1, a negative one is 0.
Now attention is paid to the samples of train data:

Table 2 Samples of train data

Number	**Review**	**Label**
1	What a hard time Robert had!	1
2	Those who are not intelligent can't stand it. I'd like you to get a refund	0
3	A movie that is worth 100%	1
4	Do not go with your children. There are sexual scenes, which I don't like	0
5	The best one of Nolan's movies. Tears come out	1
6	Actors' acting was excellent. Not boring	1
7	A story about three hours' testimony and investigation	0
8	Nolan's direction makes us excited	1
9	A movie that describes the conflict among people well	1
10	Don't tell atomic bombs just for fun	1
11	Murphy's great work that uses three hours by himself	1
12	Exploding is not nuclear bombs but Oppenheimer's mind.	1
13	Murphy's winning is confirmed	1
14	Three hours' documentary about Oppenheimer	0
15	He can't be free from his self-blame and agony	1
16	Those who did not sleep well fall asleep	0
17	Time flies like an arrow	1
18	I expected the movie too much. It seems to me that it is not good enough to see it with my family	0
19	There was no touching scene, so I fell asleep.	0
20	If you do not have the background	0

	knowledge of the movie, you don’t understand the part of it.	
21	It is only a movie, but my feelings are like I read a book.	1
22	He is a person that saved and destroyed the world. Thus, the burden was too heavy for him	1
23	Not fun. A history lecture that lacks economic value	0
24	Nolan deserves his fame	1
25	Three hours’ documentary	0
26	A movie that tortures us	0
27	Very long and boring	0
28	A movie that let us knows about Oppenheimer	1
29	If you do not have the background knowledge of the movie, then time is dragging	0
30	A movie that shows Oppenheimer’s psychological influence and agony well	1

In Table 2, we provided 30 samples (30 reviews) of train data. It is worth noticing that as exemplified in Table 2, we assigned 1 or 0 to each review of train data, depending on its meaning. While 1 is a positive one, 0 is a negative one. In the following, we provide information on the so-called tokenizer and pad sequence.

2.2. Tokenizer and pad sequence

This section is devoted to going over the so-called tokenizer and pad sequence. The so-called tokenizer refers to classifying words into the several parts of speech. Note that we got rid of 22 unnecessary case markers as stop words. On the other hand, the so-called pad sequence refers to assigning each number to each word. More specifically, we assigned each number to 320 words in descending order. Note that 0 is assigned to the rest when our criterion is met. The so-called RNN recognizes the relationships among words in a row. Put differently, it recognizes the relationships among a word and its following words so that the LSTM model can predict whether or not each review is positive. We provide 200 samples on the so-called tokenizer and pad sequence, as exemplified below:

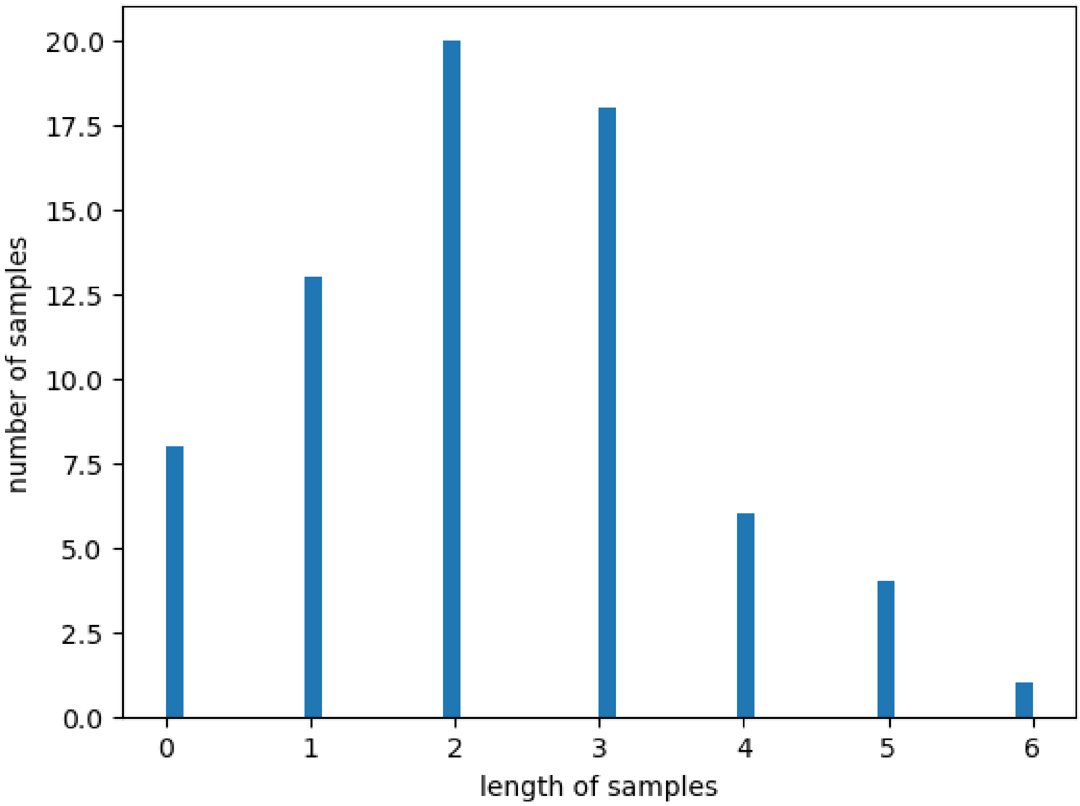
Table 3 Pad sequence

Pad Sequence
'Boring': 1, 'Movie': 2, 'See': 3, 'Too much': 4, 'Enemy': 5, '3 hours': 6, 'Thinking': 7, 'Oppenheimer': 8, 'Expecting': 9, 'Come out': 10, 'Nolan': 11, 'Acting': 12, 'No': 13, '. : 14, 'Time': 15, 'Doing': 16, ‘Direction’: 17 'Explosion': 18, 'Feeling': 19, 'Tough': 20, 'Running': 21, 'Time': 23, 'Three': ‘Receive’: 24 ‘Sleep’:25, 'Getting it': 26, 'Background': 27, '28, 'Understanding': 29, 'Great': 30, 'Good': 31: 32, 'Politics': 33, 'Da': 34, 'Audience': 35, 'Lang': 36, 'Ya': 37, 'Director': 38, 'Real': 39, 'Looking Up': 41, 'Talking': 42, 'Person': 43, 'Not': 45, 'Same': 46, 'Kill': 48, 'Murphy': 49, 'No': 50, 'Documented': 53, 'Grief': 55, 'Jul': 56, 'Kill': 58, 'Knowledge': 59, 'Content': 60, 'High': 59, 'Content': 42, 'Kill': 49, 'Murphy': 49, 'No': 53, 'Knowledge': 59, 'Content': 60, 'High': 59, 'Content':60, 'Knowledge': 59, 'It': 62, 'I want': 64, 'To': 65, 'Mann': 66, 'Mask': 67, 'Gae': 68, 'No Jam': 69, 'Disappointed': 70, 'Without': 72, 'Gi': 73, 'It's obvious': 75, 'Scene': 76, 'End': 77, 'Until': 78, 'Cha': 79, 'Expectation': 80, 'A little': 81, 'Difficult': 82, 'Robert': 83, 'How Much': 84, 'Intelligence': 86, 'Low': 87,

'People': 88, 'Human': 89, 'Reservation': 90, 'Cancel': 91, 'Recommended': 92, 'Technology': 94, '100': 95, 'Dot': 96, 'Closer': 97, 'Children': 98, 'Close': 99, 'Written': 101, 'Bust': 102, 'Why': 103, 'Tears': 104, 'Mada': 105, 'He': 106, 'Jackpot': 107, 'Medium': 109, 'Temptation': 111, 'Examination': 112, 'Always': 115, 'We': 117, 'Excited': 118, 'Combined': 119, 'Relationship': 120, 'Conflict': 121, 'Bamma': 123, 'Rodo': 124, 'Bomber': 125, 'Now': 126, 'Take it out': 127, 'Alone': 129, 'Living': 130, 'Life': 131, 'Working': 132, 'Chain': 133, 'Hack': 133, 'Certain': 149, 'Certain': 140, 'Alone': 138, 'Inside': 136, 'Heading': 136, 'Hack': 137, 'Certain': 140, 'Combined': 140, 'Daehan': 141, 'Lago': 142, 'He': 143, 'Deep': 144, 'Responsibility': 145, 'From': 147, 'Free': 147, 'The previous day': 148, 'Sleep': 149, 'Identity': 151, 'Sleeping': 152, 'Like this': 153, 'Short': 154, 'Family': 155, 'Each other': 157, 'Heavy': 158, 'Room': 159, 'Eyes': 161, 'Difficult': 164, 'Possibility': 165, 'Possible': 166, 'Book': 167, 'Reading': 169, 'World': 170, 'Saving': 171, 'I' 172, 'Crushing': 173, 'Individual': 174, 'Getting': 176, 'Life': 178, 'Weight': 178, 'Funny': 181, 'Commerciality': 182, 'Dropped': 183, 'History': 184, 'Is it okay?': 187, 'Documentary': 188, 'Pain': 189, 'Give': 191, 'Human': 192, 'Kee': 193, 'To do': 196, 'Sentence': 196, 'Psychology': 197, 'Impact': 199, 'Show': 200

Now attention is paid to the information of our data:

Figure 1 Data



It is worthwhile pointing out that as illustrated in Figure 1, the length of samples varies, depending on the number of words. Note that the number of words in a review is less than 6.

2.3. Epochs

This section focuses on probing into the loss and val_loss and the accuracy and val_accuracy with respect to train data and validation data. We trained our model to predict whether each review is positive or negative. Note that learning took place 25 times. We used a hidden layer to improve the accuracy of the LSTM model. We used sigmoid as activation and rmsprop as optimizer. Also, we used the so-called RNN that recognizes the relationships among a word and its following words. Take a look at the following table:

Table 4 Epochs

Epoch	Loss	Accuracy	Val_loss	Val_accuracy
Epoch 1	0.2447	0.9107	0.6832	0.5714
Epoch 2	0.2364	0.8929	0.4744	0.6429
Epoch 3	0.2345	0.9107	0.6777	0.6429
Epoch 4	0.2242	0.9107	0.6630	0.6429
Epoch 5	0.2169	0.9107	0.6549	0.6429
Epoch 6	0.2110	0.9107	0.6452	0.6429
Epoch 7	0.2058	0.9286	0.6326	0.6429
Epoch 8	0.2010	0.9286	0.6360	0.6429
Epoch 9	0.1967	0.9286	0.6393	0.6429
Epoch 10	0.1928	0.9286	0.6545	0.6429
Epoch 11	0.1892	0.9286	0.5984	0.6429
Epoch 12	0.1872	0.9286	0.8977	0.5714
Epoch 13	0.2025	0.8929	0.8040	0.5714
Epoch 14	0.1890	0.9107	0.7385	0.6429
Epoch 15	0.1833	0.9286	0.7295	0.6429
Epoch 16	0.1801	0.9286	0.7299	0.6429
Epoch 17	0.1773	0.9286	0.7336	0.6429
Epoch 18	0.1747	0.9286	0.7388	0.6429
Epoch 19	0.1722	0.9286	0.7454	0.6429
Epoch 20	0.1698	0.9286	0.7537	0.6429
Epoch 21	0.1675	0.9286	0.7633	0.6429
Epoch 22	0.1654	0.9286	0.7740	0.6429
Epoch 23	0.1633	0.9286	0.7854	0.6429
Epoch 24	0.1613	0.9286	0.7969	0.6429
Epoch 25	0.1593	0.9286	0.8089	0.6429

To begin with, let us define the term loss. It refers to the difference between the answer and the predictive value. Perhaps it is worthwhile noting that there was a gradual fall in the loss of train data. More specifically, there was a continual decline in the loss of train data from epoch 1 to epoch 12. Note, however, that there was a slight rise in the loss, but there was a steady decrease in the figure from epoch 14 to epoch 25. This in turn implies that our model worked well for train data. What is interesting is that there were fluctuations in the accuracy rate of our model. More specifically, the accuracy rate was 91.07% when epoch was 1. However, there was a slight fall in the accuracy rate of the LSTM model when epoch was 2. After this, there was a slight increase in the accuracy of the LSTM model in train data. Most importantly, there was no change in the accuracy rate of train data from epoch 7 to epoch 12. After this, there was a slight fall in the accuracy rate when epoch was 13. More interestingly, the LSTM model shows the same pattern from epoch 15 to epoch 25. Simply put, there was no change in the accuracy rate of the LSTM model. Most importantly, the accuracy rate of the LSTM model reached a peak when epoch was 7 (92.86%). Now let us define the term val_loss. It refers to the difference between the answer and the predictive value in validation data. It should be pointed out that the val_loss of validation data increased to 0.8977 when epoch was 12. It is worth observing that there were fluctuations in the val_loss of validation data. Note, however, that the val_loss of validation data decreased to 0.4744 when epoch was 2. It must be stressed, on the other hand, that the val_accuracy of validation data did not improve even though learning took place 25 times. It increased to 64.64%, but there was a slight decline in the val_loss of validation data when epoch was 12. That the val_accuracy of validation data was low in turn indicates that the LSTM model did not work well for validation data. We thus conclude that the LSTM model worked well for train data, but not for validation data.

Now let us turn our attention to the classification report:

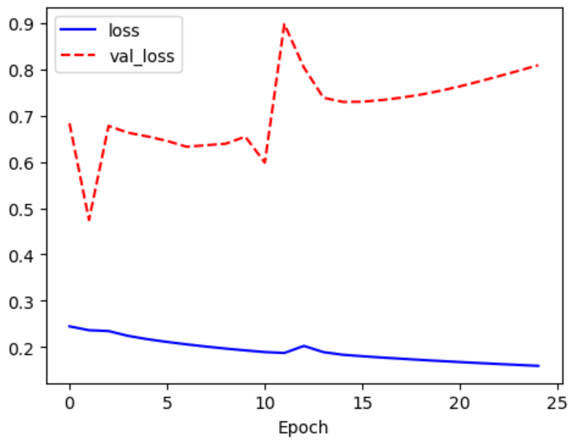
Table 5 Classification report of test data

	Precision	**Recall**	**F1-score**	**Support**
0	0.63	1.00	0.78	19
1	0.00	0.00	11	11
Accuracy			0.63	30
Macro avg	0.32	0.50	0.39	30
Weighted avg	0.40	0.63	0.49	30

The term precision means that the relevant model judged true as true. As exemplified in Table 5, the LSTM model judged 63% of 19 negative reviews as negative (63%). Notice, however, that it wrongly predicted that 11 positive reviews were not the positive ones. This in turn indicates that the LSTM model did not work properly for positive reviews in test data. Quite interestingly, as illustrated in Table 5, the accuracy rate of test data is also 63%, which is low, compared to that of train data. Finally, the term recall means the proportion of truth in what the LSTM model judged as true. As shown in Table 5, the so-called negative proportion in what our model judged as negative is 100%. On the other hand, the so-called positive proportion in what our model judged as positive is 0%. This in turn suggests that the LSTM model did not work for 11 positive reviews. Note that while we used 19 negative reviews for test data, we used 11 positive reviews. We thus conclude that the LSTM model worked for 19 negative reviews, but not for 11 positive reviews.

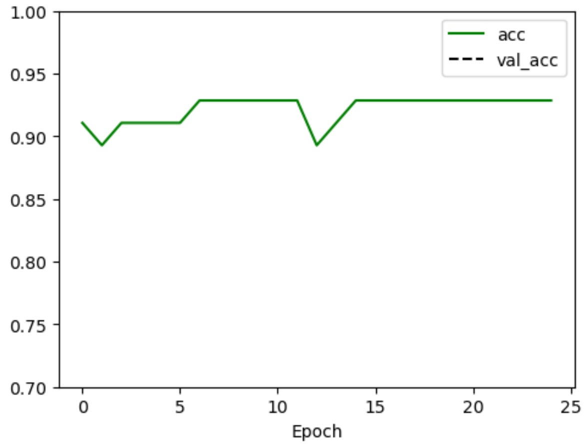
Finally, attention is paid to two graphs:

Figure 2 Loss and Val_loss



It is worth mentioning that as indicated in Figure 2, there was a steady decrease in the loss of train data. To be more specific, the loss of train data decreased to about 0.2. Notice, however, that there were fluctuations in the val_loss of validation data. More speifically, the val_loss of validation data decreased to about 0.5, but it increased to about 0.9. To sum up, the LSTM model did not work properly for validation data in that the val_loss of validation data is even higher than the loss of train data.

Figure 3 Accuracy and Val_accuracy



It is important to mention that there was a slight change in the accuracy rate of train data. Note that the accuracy rate decreased to about 89%, but it increased to about 92%. It must be emphasized, on the other hand, that the val_accuracy rate of validation data is always less than 70%. That is why the brown-dotted line disappeared in the graph. This in turn implies that the LSTM model worked properly for train data, but not for validation data. For the analysis of machine learning, deep learning, and big data, see Kang (2024a, 2024b, 2024c, 2024d, 2024e, 2024f, 2024g).

3. Conclusion

To sum up, we have trained the LSTM model to predict whether 100 reviews of the movie Oppenheimer are positive or negative. We obtained 100 reviews of the movie Oppenheimer in the portal site Naver in 2024. While we classified 70 reviews into train data, we classified 30 reviews into test data. Also, we classified 20% of train data into validation data. In section 2.3, we have argued that there was a gradual fall in the loss of train data. More specifically, there was a continual decline in the loss of train data from epoch 1 to epoch 12. There was a slight rise in the loss, but there was a steady decrease in the figure from epoch 14 to epoch 25. We have further argued that the accuracy rate of the LSTM model reached a peak when epoch was 7 (92.86%). We have also argued that the val_loss of validation data increased to 0.8977 when epoch was 12. We have maintained that the val_accuracy of validation data did not improve even though learning took place 25 times. It increased to 64.64%, but there was a slight decline in the val_loss of validation data when epoch was 12.

References

[1] Kang, N (2024a). A Big Data Analysis of a Hot Political Issue. *Studies in Linguistics* 70, 149-165

[2] Kang, N (2024b). The Ukrine War in BBC News: A Big Data Analysis. *International Journal of Applied Engineering and Technology* 6(1), 1548-1555

[3] Kang, N (2024c). A Study of the Prediction of the Model Logistic Regression in Machine Learning:

[4] Focusing on a Survey. *Semiconductor Optoelectonics* 43(1), 86-94

[5] Kang, N (2024d). The Spring of Seoul: A Big Data Analysis. *Semiconductor Optoelectonics* 43(1), 264-272

[6] Kang, N (2024e). A Study of the Prediction of the Three Models Logistic Regression, Gaussian Naive Bayes,

[7] and Multi-Layer Perceptron Classifier in Machine Learning. *Semiconductor Optoelectonics* 43(1), 253-263

[8] Kang, N (2024f). An Analysis of the Prediction Power of Four Models in Machine Learning and Deep

[9] Learning: Focusing on the Reviews of the Movie Noryang. *A Jungang Journal of English Language and Literature* 66(1), 101-122

[10] Kang, N(2024g). An Association Analysis of Shopping Items in Machine Learning. Stochastic Modelling and Computational Sciences 4(1), 97-109