Exploration of movie evaluation analysis and data preprocessing impact based on RNN technology

Ruobing Ye

Hornor Mathematics, University of Waterloo, Waterloo, N2L 3G1, Canada

r9ye@uwaterloo.ca

Abstract. In order to provide valuable models for the film and television industry, this study aims to introduce recurrent neural network (RNN) techniques for effective movie evaluation analysis. Semantic analysis using machine learning is a very important means of extracting and understanding the meaning behind the text. The study evaluates different RNN techniques to identify the optimal neural network model. Data preprocessing includes tokenization and embedding, including dataset partitioning, tokenization process and word embedding techniques. The comparative analysis involves the predictive performance of simple RNN, Long Short-Term Memory Network (LSTM) and LSTM with attention. This study also explores the impact of including emoji and punctuation analysis in the data preprocessing process on semantic analysis. The results of the study show that preprocessing emoticons and punctuation improves accuracy, and LSTM with attention shows excellent performance. Notably, the study concludes that LSTM with attention performs well in terms of runtime efficiency, convergence speed, and accuracy compared to other models. The effect of punctuation and emoticons is that it will improve the accuracy. This study helps to improve the quality of the movie by constructing an effective analytical model thus.

Keywords: Movie Evaluation Analysis, Recurrent Neural Network, Data Preprocessing, LSTM, LSTM with Attention.

1. Introduction

In today's digital age, the proliferation of online platforms and social media has empowered individuals to easily share their opinions and thoughts about movies. As a result, there has been an exponential increase in user-generated movie reviews across popular websites such as IMDb, Rotten Tomatoes, YouTube, and Amazon. These reviews have become a valuable source of information for both filmmakers and movie enthusiasts, offering real-time feedback and insights into the audience's reactions to films. By understanding the public's sentiments towards specific movies, Machine Learning (ML)-driven movie review analysis helps the film industry make data-driven decisions and improves the movie-watching experience worldwide.

There are numerous studies already exists in the field of text semantics analysis and each of them focus on different aspects. Many of these studies uses data from these public social media platforms, such as Twitter data, news data, product review data and etc [1-3]. Additionally, there are also studies has investigated new model. There is one report focus on large-scale text analysis and introduce a new model proposes a novel multimodal emotion analysis model called Multi-view Attentional Network

(MVAN) [4]. Furthermore, data preprocessing holds significant importance within this area and there is report has discussed the importance of it [5]. Various approaches are available, for instance, one the studies investigate how to map each word with a number of scales to justify the positive and negative during the data pre-processing stage and using the scale to perform the semantic analysis [6]. Also, another paper implements an algorithm that assigns weights to the scores of both the hashtag and the processed text, aiming to derive the overall sentiment [7]. Textual semantic analysis is not limited to the English language, there are also reports that studies on other languages, such as, Chinese and French and even multilingual contexts [8-10]. While these studies share many data preprocessing steps, they also incorporate distinct techniques tailored to each language's characteristics.

The main objective of this study is to introduce the Recurrent Neural Network (RNN) technique to construct a model to analyze the movie evaluation effectively. The study compares different RNN techniques to find the most effective neural network model. In addition, data preprocessing is performed by introducing tokenization and embedding techniques. Specifically, first, the dataset is evenly divided into half training set and half testing set [11]. Second, since it is difficult to apply the model directly to text, a series of tokenization steps are used in the data preprocessing stage and embedding techniques are utilized to map word tokens to points on a multidimensional vector space. Third, the prediction performance of different models such as simple RNN, Long and Short-Term Memory Network (LSTM) and LSTM with attention is analyzed and compared. In addition, the impact of merging emoji and punctuation analyses at the data preprocessing stage is analyzed. This can help optimize the results and investigate whether they contribute to the semantic analysis of the text. The experimental results show that preprocessing emoticons and punctuation improves accuracy and that LSTM with attention has the best performance. This study can provide valuable models to the film and television industry thus improving the quality of movies.

2. Methodology

2.1. *Dataset description*

This paper focused on IMDb movie reviews dataset for sentiment analysis. The dataset [11] includes movie reviews and their corresponding binary sentiment polarity labels. Its primary purpose is to act as a reference for sentiment classification evaluations. This document provides insights into the data collection methodology and offers guidance on utilizing the provided files effectively. As shown in table 1, this dataset contains a total of 50,000 movie reviews, which are divided into 25,000 samples for training and 25,000 samples for testing. Notably, the data set maintained a balanced category distribution, with 25,000 positive comments and the same number of 25,000 negative comments. This balanced dataset is essential for unbiased model training and reliable sentiment classification.

Table 1. Dataset Summary.

	Positive	Negative
Training	12,500	12,500
Testing	12,500	12,500

2.2. *Proposed approach*

2.2.1. Dataset pre-processing. Each movie review in the dataset is a variable-length text sequence, often containing multiple sentences. These reviews are raw and unprocessed, meaning they may encompass diverse types of text and language styles. Preprocessing is necessary to clean and prepare the data for analysis. According to the steps shown in the figure 1, data pre-processing steps are divided into 2 parts: tokenization and embedding.

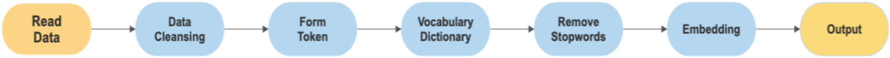

Figure 1. Data Pre-processing Flowchart.

In the field of natural language processing (NLP), tokenization is a fundamental preprocessing step to break down a sentence into individual words or sub words, known as tokens. With the re library in Python, constructing a lexer tokenizer is a very straightforward process with several essential operations. It starts with a few steps of data cleansing. Firstly, removing the HTML tokens, this will ensure that any HTML tags or elements present are eliminated, leaving only the relevant text. Another common practice in NLP to avoid issues related to case sensitivity. Converting all text to lowercase can ensure that the tokens are uniform and do not differentiate between uppercase and lowercase letters. Similarly, taking care of tense and plural is also an important step. Removing them will reduce different word forms to their common base or root form. For example, “went, goes → go”, and “books → book”. This can help reduce the size of the vocabulary, decrease data sparsity, and extract shared sentiment information. One last step before generating the word token is handling the emoji and punctuation. The impact of processing them is also one of the main focuses of this study. Emoji tokens are an essential part of modern text communication and extracting them as separate tokens can be beneficial. Placing the emoji tokens at the end of the sentence ensures that they are preserved but do not interfere with the regular word tokens’ semantics. Punctuation marks, such as commas, periods, and exclamation marks, are often irrelevant in many NLP tasks. Thus, removing them during tokenization helps streamline the process and create cleaner token sequences.

After finishing the data cleansing process, the next step for data pre-processing is generating word tokens. Using the tokenizer to generate word tokens, which are the main components of the sentence used for further analysis and processing. For example, we have an IMDb comment, “This is awesome movie <br /><br />. I loved it so much :-) I’m goona watch it again :)”, and it will generate tokens: [‘this’, ‘is’, ‘awesome’, ‘movie’, ‘i’, ‘loved’, ‘it’, ‘so’, ‘much’, ‘i’, ‘m’, ‘goona’, ‘watch’, it’, ‘again’, ‘:)’, ‘:)’]. Then, counting the frequency of each word tokens and arranging them as a vocabulary dictionary from high frequency to low frequency. In addition, check through the vocabulary dictionary, there exists some common stopwords that appear frequently but often do not carry much emotional or meaningful information, such as “a”, “the”, “is”, and so on. removing these can reduce the dimensionality of the feature space and improve the efficiency of the model.

At the end, it comes to the embedding stage, which plays a key role in converting high-dimensional discrete categorical data (such as words) into low-dimensional continuous vector space. This technique allows to represent words as dense real-valued vectors, capturing semantic relationships and similarities between them. In this case, similar words are mapped to adjacent points in the vector space.

2.2.2. Simple RNN model. RNN is a type of neural network commonly used to analyze text semantics. Unlike traditional neural networks, RNNs possess connections that loop back on themselves, enabling them to capture temporal dependencies within sequences. Firstly, input text data word by word or in chunks, and then the network will learn from the patterns in the sequence. At each time step, an RNN takes a word from input and combines it with the hidden state from the previous step, and an output is generated, and it will update its internal state. This allows RNN to comprehend the semantic significance of text and interpret context across time and makes them highly efficient for sentiment analysis. It is easy to implement and suitable for sequential data, such as text. So, it is a reasonable starting point for analyzing text semantics.

2.2.3. LSTM and LSTM with Attention models. Since the traditional RNN model is facing vanishing gradient problem, another model LSTM, which is advanced type of RNN, is developed to address it. It incorporates a sophisticated gating mechanism which includes 3 different essential gates: the forget gate, the input gate, and the output gate. These gates enable LSTM networks to regulate the flow of information over time, enhancing their ability to capture long-term dependencies in sequences. This can cover the gradient vanishing and gradient explosion challenge that RNN was facing.

LSTM can also combine with attention mechanism. Because its algorithm is based on human cognitive processes, attention mechanism allows it to allocate its focus to specific parts of the text like human beings. By assigning varying levels of importance to different words or phrases, the network can effectively capture subtle relationships and differences within the text. Moreover, comparing to basic LSTM, introducing the attention mechanism can selectively concentrate on relevant sections, mitigating the risk of information loss, which will increase the accuracy to out semantic analysis.

2.2.4. Loss function. The network is constructed under a binary cross entropy loss function. There are only two classes of loss, 0 and 1, where 0 represent negative semantic and 1 represent positive semantic. The loss function is designed to calculated at the end at the output stage. This approach allows the network to effectively optimize its prediction accuracy based on sentiment classification.

3. Result and discussion

After the data-preprocessing steps, 3 distinct neural networks, RNN, LSTM, and LSTM with attention have been applied to the dataset for semantic analysis. This report focuses on the ability to analyze large and complex dataset and tries to find the one that has the best performance among these three networks. This evaluation is based on the accuracy and time-consumption. Furthermore, to investigate the impact of punctuation and emoji on semantic analysis, removing step 3 and 4 during the data pre-processing stage. Then, perform one of the neural networks that has the best performance and conduct a comparative analysis between the modified dataset and the original dataset.

3.1. Simple RNN Model

From table 2, all three sets begin with a modest accuracy of merely 60%, then shows a gradual improvement in the later epoch. The training set ends up with over 90% accuracy, whereas both validation and testing set only achieve 75% accuracy. The increasing trending shown in figure 2 suggests that the model keeps learning from the training data. Nevertheless, there exists some fluctuations in the training process, and the accuracy for validation and testing sets share a similar trend with many overlaps. This implies that there might be some overtraining on dataset.

Table 2. Training Result for Simple RNN Model.

Epoch	**Accuracy**		
	Training Set	**Validation Set**	**Testing Set**
1	0.5918	0.598	0.5937
2	0.6917	0.6894	0.6913
3	0.7629	0.6912	0.7055
4	0.7889	0.6912	0.7055
5	0.8256	0.7202	0.7358
6	0.8506	0.7684	0.772
7	0.8451	0.7414	0.7512
8	0.8796	0.7684	0.7743
9	0.8999	0.7560	0.7633
10	0.9176	0.7504	0.7565
11	0.8727	0.7488	0.7529
12	0.9233	0.7594	0.7645

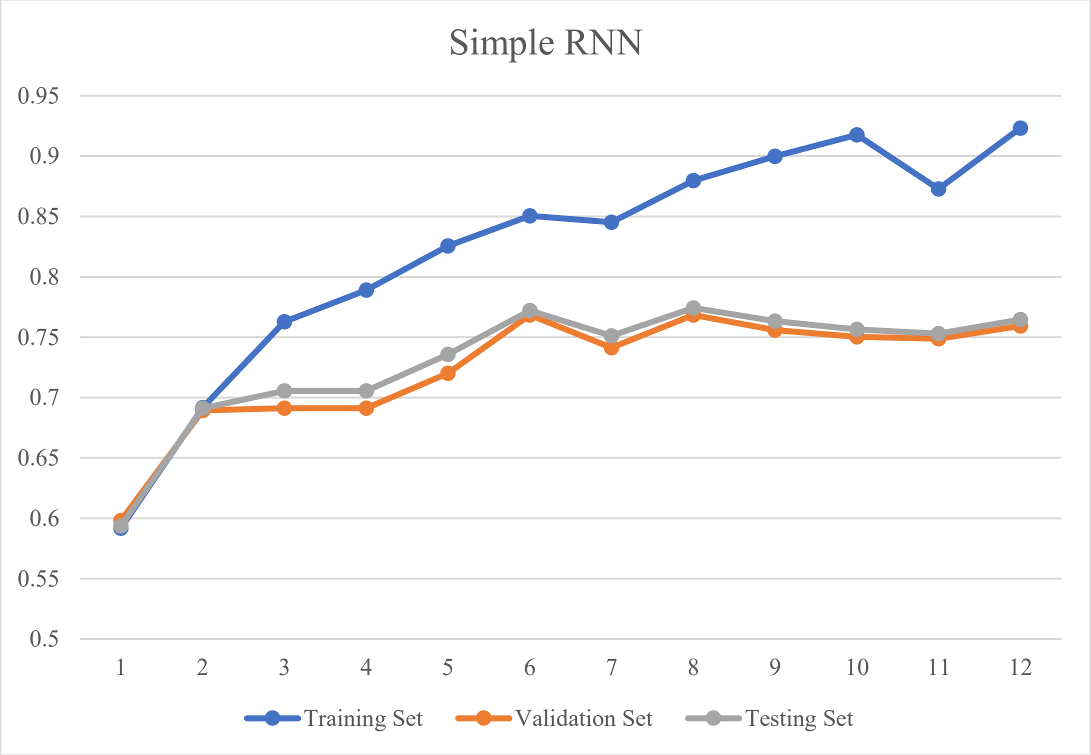

Figure 2. Accuracy on Training, Validation and Testing Sets for Simple RNN.

3.2. LSTM Model

It is easy to observe from both table 3 and figure 3 that the accuracy of epoch 12 is much higher than epoch 1 for all three sets. Therefore, the model's accuracy trend increases over the initial epochs to the later epochs. However, it is important to note that there is a drop for validation and testing set at epoch 5, which means that it is not consistently increasing with each epoch. This indicates that the LSTM model may have over-fitting the training data.

Table 3. Training Result for LSTM Model.

Epoch	**Accuracy**		
	Training Set	**Validation Set**	**Testing Set**
1	0.7431	0.7102	0.6933
2	0.8219	0.8090	0.7888
3	0.8774	0.8394	0.8142
4	0.9095	0.8236	0.7971
5	0.9364	0.7954	0.7541
6	0.9463	0.8414	0.8158
7	0.9611	0.8398	0.8097
8	0.9665	0.8422	0.8062
9	0.9787	0.8360	0.8053
10	0.9781	0.8410	0.8116
11	0.9876	0.8458	0.8118
12	0.9915	0.8344	0.7973

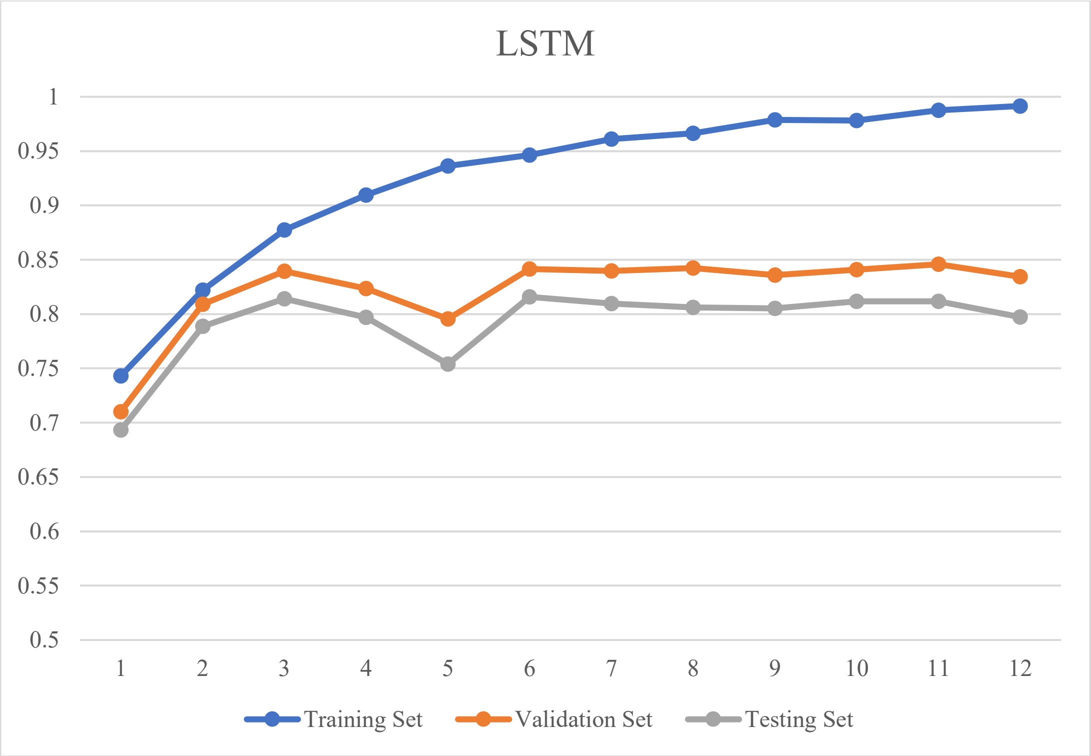

Figure 3. Accuracy on Training, Validation and Testing Sets for LSTM.

3.3. LSTM with Attention Model

As shown in table 4, the training set, starts with around 75%, but increases rapidly to almost 100% at epoch 4 and remain until epoch 13. The accuracy for both validation and testing sets stays around 87% to 90%. Overall, observing the trends in figure 4, the LSTM with attention's accuracy remains consistently high and starts to become very stable after epoch 4. This can conclude that LSTM with attention model is effectively learning the underlying patterns in the data and not overfitting to the training set.

Table 4. Training Result for LSTM with Attention Model.

Epoch	Accuracy		
	Training Set	Validation Set	Testing Set
1	0.7568	0.8800	0.8700
2	0.9182	0.8976	0.8856
3	0.9638	0.8704	0.8634
4	0.9869	0.8890	0.8782
5	0.9950	0.8920	0.8795
6	0.9943	0.8900	0.8748
7	0.9967	0.8856	0.8725
8	0.9959	0.8926	0.8762
9	0.9997	0.8806	0.8526
10	0.9979	0.8908	0.8752
11	0.9985	0.8826	0.8694
12	0.9976	0.8862	0.8691

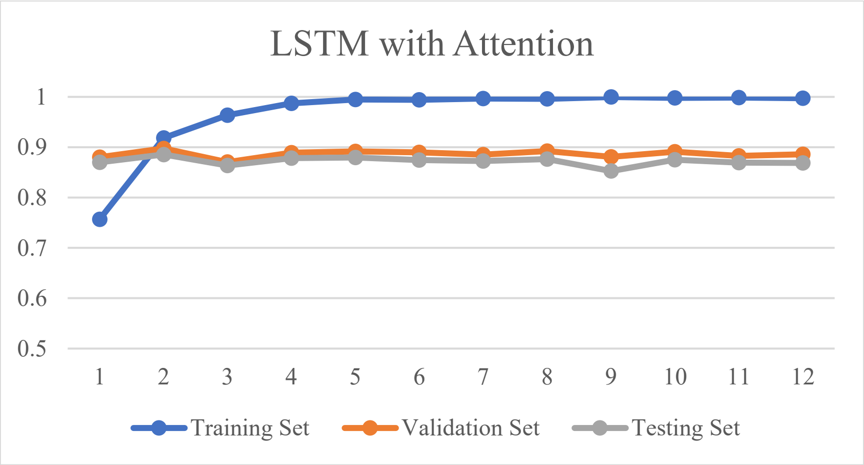

Figure 4. Accuracy on Training, Validation and Testing Sets for LSTM with Attention.

3.4. Comparison between Simple RNN, LSTM and LSTM with Attention Models

Figure 5 contains the accuracy for testing set for all 3 models. It is easy to conclude that for most of epochs, the LSTM with attention consistently demonstrates a high accuracy compared to the other two models. The LSTM model generally achieves higher accuracy levels than the RNN model. Across the range of epochs, all three models show an increasing trend, interspersed with fluctuations. Notably, the LSTM with attention initiates with most stable trend and sustains its lead over the other models across most epochs. Although the LSTM model and the LSTM with attention model exhibit analogous accuracy trends, the latter maintains a marginal advantage. In contrast, the RNN model consistently trails behind the other two models in terms of accuracy. Moreover, the processing time for RNN is much longer than the other 2 models. This is because the data transmission within the network involves several multiplication operations through the activation function. This can cause the gradient to exponentially rocket up or fall over time, which potentially resulting in an unstable training process. Therefore, simple RNN encounters challenges like gradient vanishing and gradient explosion, and it is hard to make improvement. Overall, the LSTM with attention emerges as the top performer among the three models in terms of accuracy, followed by the LSTM, and subsequently the RNN.

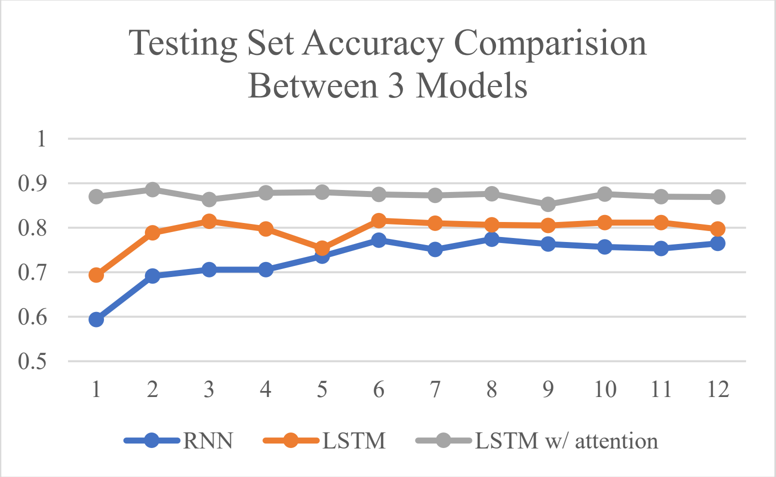

Figure 5. Testing Sets Accuracy for RNN, LSTM and LSTM with Attention Models.

3.5. Impact of Punctuation and Emoji in Data Pre-Processing Stage

It is worth noting from figure 6 that, compared with the accuracy without punctuation and emoticon, the accuracy with punctuation and emoticon is consistently a little higher under the same model. The variations in accuracy between the two sets of data are relatively small, with some minor differences observed. Therefore, the impact of punctuation and emoji is not significantly pronounced but including them in the data pre-processing stage will certainly improve the accuracy of semantic analysis.

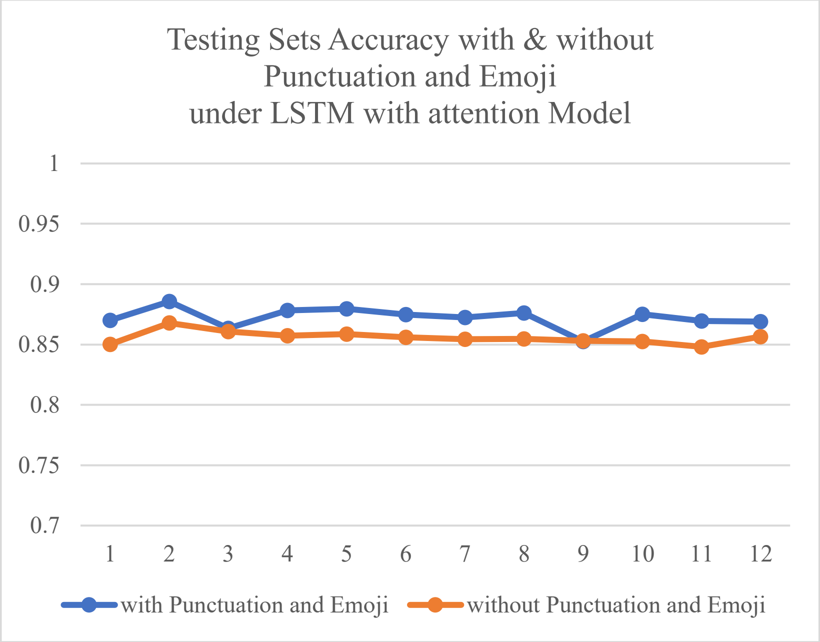

Figure 6. Testing Sets Accuracy with & without Punctuation and Emoji under LSTM with attention.

4. Conclusion

This study introduces the deep learning method to construct model while comparing the difference between conducting data pre-processing with and without punctuation and emoji to study their impact on semantic analysis. The results shows that although the impact is relatively modest in this case, incorporating them as one of the elements that effect the semantics can increase the accuracy. As the dataset scales up in real world, the potential impact might become significant. Moreover, it used various neural network models, RNN, LSTM and LSTM with attention, to conduct meticulous sentiment analysis experiment and compute a thorough evaluation between them. It concludes that LSTM with attention has superior runtime, convergence speed, and highest accuracy among three of them. It also has the ability to intelligently focus on relevant aspects of textual content. This result highlights the effectiveness of attention mechanisms in boosting the efficiency and accuracy of sentiment analysis procedures. In the future study, in order to having improvement on semantic analysis tools, data pre-processing needs deeper investigation. Having a data processor that can yield cleaner and more logically structured data inputs will increase the efficiency. Additionally, optimizing LSTM with Attention's runtime holds immense potential. Continuing improving the attention mechanism, aiming to bolster both the model's accuracy and processing speed.

References

[1] Agarwal A Xie B Vovsha I et al. Sentiment analysis of twitter data//Proceedings of the workshop on language in social media (LSM 2011) 2011: pp 30-38

[2] Alexandra B Ralf S Mijail K 2010 Sentiment Analysis in the News Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC'2010) pp 2216-2220

[3] Fang X Zhan J 2015 Sentiment analysis using product review data Journal of Big Data 2: p 5

[4] Yang X Feng S Wang D et al. 2020 Image-text multimodal emotion classification via multi-view attentional network IEEE Transactions on Multimedia 23: pp 4014-4026

[5] Haddi E Liu X Shi Y 2013 The role of text pre-processing in sentiment analysis Procedia Comput Sci 17 pp 26-32

[6] Taboada M 2016 Sentiment analysis: An overview from linguistics Annual Review of Linguistics 2: pp 325-347

[7] S. Pradha S Halgamuge M N Vinh N T Q 2019 Effective text data preprocessing technique for sentiment analysis in social media data//2019 11th international conference on knowledge and systems engineering (KSE) IEEE pp 1-8

[8] Tan S Zhang J 2008 An empirical study of sentiment analysis for chinese documents Expert Systems with Applications 34(4): pp 2622-2629

[9] Apidianaki M Tannier X Richart C 2016 Datasets for Aspect-Based Sentiment Analysis in French Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16) pp 1122–1126

[10] Dashtipour K Poria S Hussain A et al. 2016 Multilingual Sentiment Analysis: State of the Art and Independent Comparison of Techniques Cogn Comput 8 pp 757–771

[11] Maas A Daly R E Pham P T et al 2011 Learning word vectors for sentiment analysis Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies pp 142-150