Sentiment Analysis with Different Deep Learning Methods

Zixiang Chen
School of Software Engineering, Chongqing University of Post and Telecommunications, Chongqing, 400065, China

Keywords: CNN, RNN, LSTM, Sentiment Analysis, Deep Learning

Abstract: Sentiment analysis is a task of natural language processing that seeks to identify and produce the feelings or viewpoints conveyed in written or spoken communication. It has various applications, such as social media analysis, product reviews, customer service, chatbots, recommender systems, etc. In this paper, the author evaluates the method on a large-scale dataset of fine foods reviews from Amazon, and compares it with several models namely CNN, RNN and LSTM. The paper evaluates the outcomes of each model on three metrics: Recall, Precision and F1 Score. The findings indicate that LSTM outperforms both TextCNN and RNN across all metrics, making it the most effective model for this task. The paper also discusses the possible reasons for the superiority of LSTM, such as its capacity to record context and long-term dependencies. The paper also analyzes the advantages and disadvantages of TextCNN and RNN, such as their speed, simplicity, and robustness. The paper provides empirical evidence for the effectiveness of different models for sentiment analysis.

1 INTRODUCTION

Sentiment analysis is a research area that aims to identify or generate the emotional attitude, mood or tendency of natural language texts or speeches. Sentiment analysis can be applied for many different situations. For example, product reviews, chatbots, recommender systems, etc. Sentiment analysis can help users and businesses to understand the opinions, preferences and feedbacks of customers or users, and provide better products or services (Nandwani & Verma 2021, Le-Khac et al. 2020, Sejwal et al. 2021).

Sentiment analysis is a challenging task, as it involves various aspects of NLP (natural language processing), such as syntactic analysis, lexical analysis, semantic analysis, pragmatic analysis, etc. Moreover, sentiment analysis is influenced by many factors, such as the context, the domain, the culture, the subjectivity, the sarcasm, the irony, etc. Therefore, sentiment analysis requires not only the understanding of the literal meaning of the texts or speeches, but also the inference of the implicit meaning and the emotional expression.

However, most of the existing deep learning methods for sentiment analysis are embedded in supervised learning, which requires a large amount of labeled information for training and testing (Kohsasih et al. 2022). The labeling process is often time-consuming, labor-intensive, and subjective, and the labeled data may not cover all the possible scenarios and domains of sentiment analysis (Li et al. 2020, Bordoloi & Biswas 2023). Moreover, the supervised learning methods may suffer from the problems of overfitting, data imbalance, domain adaptation, cross-lingual transfer, etc (Zhao et al. 2021, Liu et al. 2020).

This paper proposes an unsupervised deep learning method for sentiment analysis. It uses contrastive learning with CNNs, RNNs, and LSTMs to learn sentiment representations from texts or speeches without labels. This method can handle various sentiment analysis tasks, such as classification, similarity, and generation.

To evaluate the method, the paper uses a large-scale dataset of online foods comments from Amazon, including more than 500,000 reviews.

2 METHODS

In this section, the paper describes the methods that are used for sentiment analysis based on unsupervised learning. The paper first introduces the contrastive learning framework. Then, the paper describe the encoder models we used to encode the texts or

Chen, Z.
Sentiment Analysis with Different Deep Learning Methods.
DOI: 10.5220/0012832800004547

In *Proceedings of the 1st International Conference on Data Science and Engineering (ICDSE 2024)*, pages 154-159
ISBN: 978-989-758-690-3

speeches into latent vectors, including CNNs, RNNs, and LSTMs. Next, the paper introduces the self-supervised auxiliary task that are used to enhance the sentiment representations.

2.1 Contrastive Learning Framework

Contrastive learning is a particular kind of unsupervised learning that learns the representations of data through increasing the positive pairings' agreement and decreasing the negative pairs' agreement. The positive pairs are the data that have the same or similar labels, while the negative pairs are the data that have different or opposite labels. In this case, the labels are the sentiments of the comments, such as positive, negative, or neutral. The intuition behind contrastive learning is that the data with the same or similar sentiments should have similar representations, while the data with different or opposite sentiments should have dissimilar representations.

To implement contrastive learning, we need to define a contrastive loss function in order to measure the agreement between the pairs of data. There are different types of contrastive loss functions, such as triplet loss, InfoNCE loss, and NT-Xent loss. In this paper, we use the NT-Xent loss. And the NT-Xent loss is defined as follows:

$$L_{NT-Xent} = -\frac{1}{2N}\sum_{i=1}^{N}\left[log\frac{exp(sim(z_i,z_{i^+})/\tau)}{\sum_{j=1}^{2N} 1_{[j\neq i]}exp(sim(z_i,z_j)/\tau)} + log\frac{exp(sim(z_{i^+},z_i)/\tau)}{\sum_{j=1}^{2N} 1_{[j\neq i^+]}exp(sim(z_{i^+},z_j)/\tau)}\right] \quad (1)$$

Where N is the batch size, z_i and z_{i^+} are the positive pair's latent vectors, sim() stands for the cosine similarity, τ represents a temperature parameter. Positive pairs are encouraged to have high similarity whereas negative pairs are encouraged to have low similarity by the NT-Xent loss, and thus learns the sentiment representations in an unsupervised manner.

2.2 Models

To encode the comments into latent vectors, the paper use three types of encoder models, namely CNNs, RNNs, and LSTMs. These models are widely applied in NLP and have demonstrated strong performance in different jobs.For example, machine translation, sentiment analysis, and text categorization.

2.2.1 CNN

CNNs are composed of pooling layers, convolutional layers and fully connected layers. Pooling layers reduce the dimensionality of the data and retain the most important information. Convolutional layers apply filters to the input data and extract local features. Fully connected layers combine the features and produce the output. CNNs can capture the n-gram features of the texts or speeches and learn the hierarchical representations.

The convolutional layers are formulated as follows:

$$h_i = f(W * x_i + b) \quad (2)$$

where x_i represents the input data, W is the convolutional filter, b is the bias term, $*$ is the convolution operation, f is the activation function, and h_i is the output feature map.

The pooling layer can be formulated as follows:

$$p_i = g(h_i) \quad (3)$$

g is the pooling function, such as max, average, or sum. The fully connected layer is formulated as follows:

$$o_i = f(Wp_i + b) \quad (4)$$

where p_i is the input pooled feature map, and o_i is the output vector.

Figure 1 illustrates the TextCNN Architecture, the neural network for the classification of texts. The model takes an input text and transforms it into numerical data through an embedding layer. After that, two parallel convolutional layers and max-pooling layers handle the embedded input in order to extract and improve important textual properties. The extracted features are concatenated and passed through a dropout layer to prevent overfitting. Batch normalization is applied to standardize the inputs, promoting model training efficiency and stability. Finally, a dense layer feeds into a softmax function that classifies the input text into appropriate categories, producing the final output.

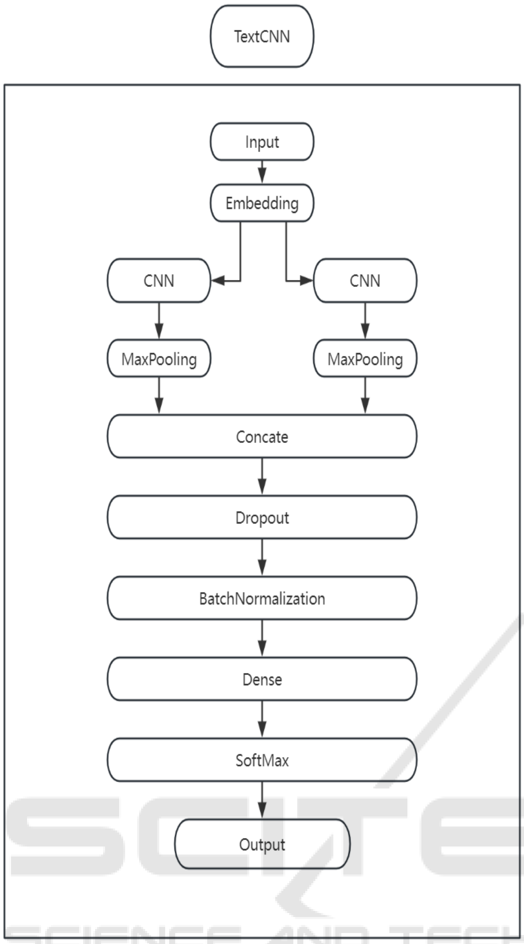

Figure 1: TextCNN Architecture (Picture credit: Original).

2.2.2 RNN

RNNs are composed of recurrent units, such as LSTM or GRU. RNNs can capture the long-term dependencies and temporal dynamics of the texts or speeches and learn the sequential representations.

The recurrent unit can be formulated as follows:

$$H_i = f(W_x x_i + W_h h_{i-1} + b) \quad (5)$$

H_i represents the hidden state, W_x and W_h are weight matrices

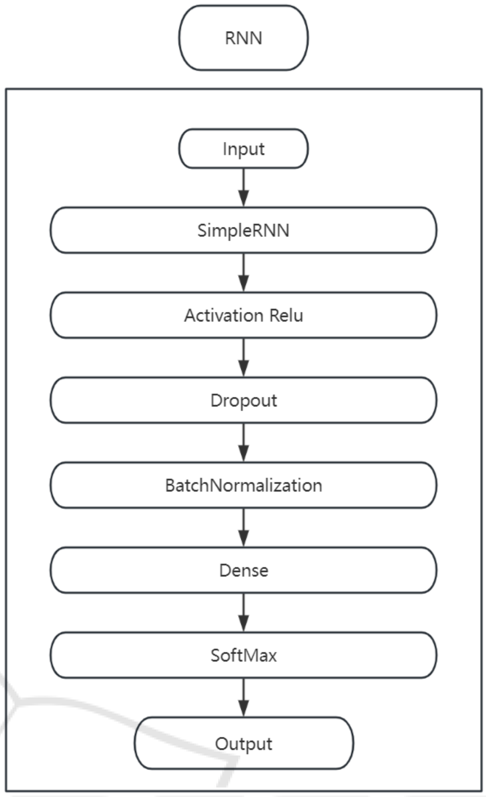

Figure 2: RNN Architecture (Picture credit: Original).

Figure 2 illustrates the RNN Architecture, a particular kind of neural network designed for processing sequential data. This model takes an input data and passes it through a series of layers, each performing a specific function essential for the training and operation of the neural network (Liu et al. 2020, Wang et al. 2022). The data's temporal patterns and sequences are processed by the SimpleRNN layer. To facilitate the learning of intricate patterns, non-linearity is implemented by the Activation Relu layer. The Dropout layer is used for regularization to prevent overfitting. The Batch Normalization layer helps in faster and more stable training, while the Dense layer is used for output generation. The final SoftMax layer classifies the outputs into various categories, producing the final output.

2.2.3 LSTMs

LSTM consists of recurrent units that handle the input data in a sequential manner while keeping track of a hidden state that contains the data from earlier inputs (Yuan et al. 2020). LSTM can capture the long-term dependencies and temporal features of text or speech and learn the sequence representation.

The recurrent unit can be expressed as:

$$H_i = f(W_x x_i + W_h h_{i-1} + b) \tag{6}$$

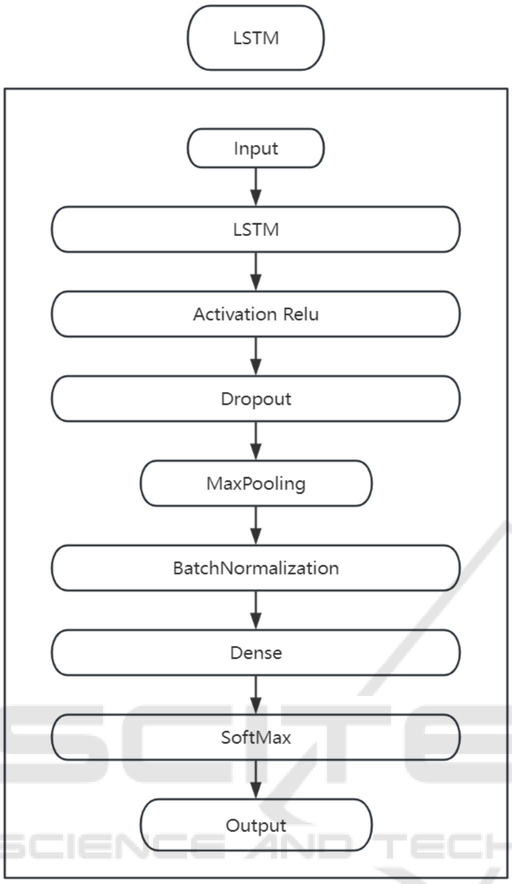

Figure 3: LSTM Architecture (Picture credit: Original).

Figure 3 illustrates the LSTM Architecture, a particular kind of neural network designed for processing sequential data. The model takes an input data and passes it through a series of layers, each performing a specific function essential for the training and operation of the neural network. Processing sequences and temporal patterns in the data, as well as upholding a memory state that can store and retrieve pertinent information over extended periods of time, are the responsibilities of the LSTM layer. The model gains non-linearity from the Activation Relu layer, which improves learning. The Dropout layer is used for regularization to prevent overfitting. The MaxPooling layer reduces the output volume's dimensions in space, and highlights the dominant features. The Batch Normalization layer normalizes the activations of the neurons, improving generalization and speeding up training. The Dense layer is used for learning features and making predictions, and finally, the SoftMax layer is used at the output end to provide probabilities for each class in multi-class classification tasks.

2.2.4 Self-Supervised Auxiliary Task

To enhance the sentiment representations learned by contrastive learning, the paper uses a self-supervised auxiliary task, which is a task that does not require any human annotation and can generate labels from the data itself. The self-supervised auxiliary task the paper use is the MLM (masked language modeling) task. The MLM task masks some tokens randomly in the input data, considers the context and anticipates the original tokens. The MLM task can improve the semantic and emotional information of the latent vectors and make them more informative and diverse.

3 RESULTS

In this part, the paper shows the outcomes of the experiments on sentiment analysis using different machine learning models. The paper uses 3 typical models and compare each of the results: CNN, RNN, and LSTM. The paper uses these metrics to evaluate how well the models can correctly classify the sentiment of the comments.

3.1 Datasets

To obtain the texts or speeches for sentiment analysis, the paper uses a large-scale dataset of online foods reviews from Amazon, which covers a duration of over ten years, encompassing all 568,454 reviews completed up to October 2012. This dataset is suitable for this method, as it contains rich and diverse texts and speeches with various sentiments, and it does not require any manual labeling.

To process the data, we first filter out the reviews that are too short or too long, and keep the reviews that have between 50 and 500 words. Then, we convert the ratings into 3 sentiment labels, namely positive, neutral, and negative based on the following rules:

If the rating is 4 or 5 stars, the sentiment label is positive.

If the rating is 1 or 2 stars, the sentiment label is negative.

If the rating is 3 stars, the sentiment label is neutral.

The paper uses the sentiment labels to form the positive and negative pairs for contrastive learning, and to evaluate the performance of this method oan sentiment classification. We also use the sentiment labels to mask some tokens in the reviews for the MLM task.

3.2 Data Collection and Processing

This paper preprocessed the text data by removing punctuation, stop words, numbers, URLs, etc., and performing tokenization, stemming, part-of-speech tagging, etc. The paper used word embedding techniques called Word2Vec to convert each word into a fixed-length vector, as the input of the models.

The author trained and tested each model on the same dataset, using an 80/20 split for training and testing. The paper used the same hyperparameters for each model, such as batch size (32), epochs (15), etc. The paper used the NT-Xent loss function to optimize the models, as it promotes high similarity between positive pairs and low similarity between negative pairs, thus learning sentiment representations in an unsupervised way.

3.3 Analysis

The outcomes of 3 models are shown in the bar chart below, which compares the p, r, f score of the three models on both datasets.

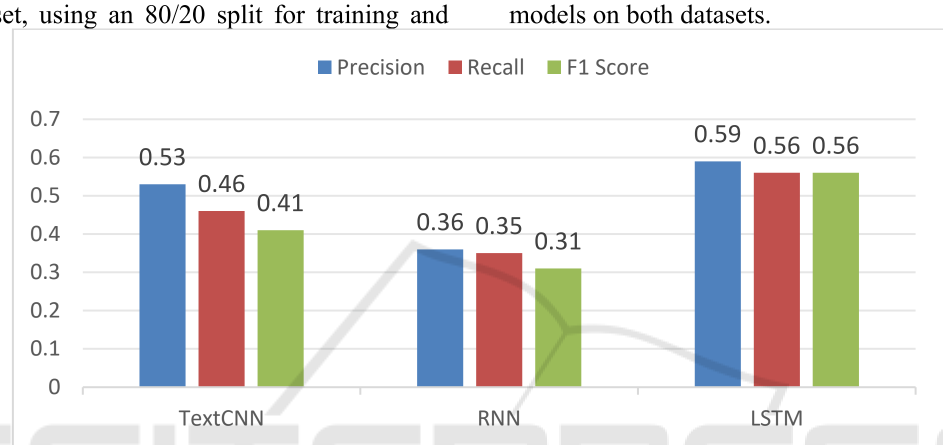

Figure 4: Result comparison.

As can be seen from Fig.4, TextCNN exhibits moderate performance with a precision of 0.53 indicating that over half of the positive classifications were accurate. However, its recall of 0.46 signifies that it missed a significant portion of actual positive instances leading to a lower F1 score. TextCNN is fast and simple to implement, but it may not capture the global context or long-term dependencies of the text. This is because TextCNN's filters have a fixed size, which means that they can only cover a part of the text, not the whole text. This implies that TextCNN may miss important information in the text, or fail to understand the overall meaning of the text.

RNN has the lowest performance among the three models with all metrics below 0.4; this could be attributed to its difficulty in handling long-range dependencies or capturing semantic meanings from the reviews. RNN is a recurrent neural network that processes the text sequentially and updates its hidden layer every step. It can model the temporal dynamics of the text, but it suffers from the vanishing or exploding gradient problem, which makes it hard to learn from distant information.

LSTM outperforms both TextCNN and RNN across all metrics making it the most effective model for this sentiment analysis task; its architecture allows it to capture long-term dependencies effectively and understand context better leading to more accurate predictions.

4 CONCLUSION

In this paper, the author has presented a comparative study of three different models - CNN, RNN, and LSTM - for sentiment analysis based on Amazon food reviews. The paper has evaluated the outcomes of each model on 3 criteria. The results demonstrate that LSTM outperforms both CNN and RNN across all metrics, making it the most effective model for this task. The paper has also discussed the possible reasons for the superiority of LSTM, such as its ability to highlight long-term reliance and context information.

This study provides empirical evidence for the effectiveness of LSTM for sentiment analysis. It also has practical implications for applications that rely on sentiment analysis, such as recommender systems, customer service, and social media analysis. By using LSTM, these applications can achieve higher accuracy and reliability in detecting and analyzing the sentiments of users.

However, the study also has some limitations that suggest directions for future research. First, the paper has only used one dataset of Amazon food reviews, which may not be representative of other domains or genres of text. Therefore, it would be interesting to test the generalizability of the findings on other datasets, such as movie reviews, product reviews, or tweets. Second, the paper has only compared three models, which may not cover the full spectrum of possible models for sentiment analysis. Therefore, it would be worthwhile to explore other models, such as attention-based models, transformer models, or graph neural networks, and compare their performance with LSTM.

In conclusion, the paper has demonstrated that LSTM is a powerful and robust model for sentiment analysis based on Amazon food reviews. Hope that the study can inspire further research on this topic and provide useful insights for practitioners and developers who want to leverage sentiment analysis in their applications.

REFERENCES

Bordoloi, M., Biswas, S.K. Sentiment analysis: A survey on design framework, applications and future scopes. *Artif Intell Rev* 56, 12505–12560 (2023).

G. Li, Q. Zheng, L. Zhang, S. Guo and L. Niu, "Sentiment Infomation based Model For Chinese text Sentiment Analysis," *2020 IEEE 3rd International Conference on Automation, Electronics and Electrical Engineering* Shenyang, China, 366-371 (2020).

H. Liu, I. Chatterjee, M. Zhou, X. S. Lu and A. Abusorrah, "Aspect-Based Sentiment Analysis: A Survey of Deep Learning Methods," in *IEEE Transactions on Computational Social Systems*, vol. 7, no. 6, pp. 1358-1375. (2020).

K. L. Kohsasih, B. H. Hayadi, Robet, C. Juliandy, O. Pribadi and Andi, "Sentiment Analysis for Financial News Using RNN-LSTM Network," 2022 4th International Conference on Cybernetics and Intelligent System (ICORIS), Prapat, Indonesia, 1-6, (2022)

Nandwani, P., Verma, R. A review on sentiment analysis and emotion detection from text. *Soc. Netw. Anal. Min.* 11, 81 (2021).

P. H. Le-Khac, G. Healy and A. F. Smeaton, "Contrastive Representation Learning: A Framework and Review," in *IEEE Access*, vol. 8, pp. 193907-193934, (2020).

S. Sejwal, N. Faujdar and S. Saraswat, "Sentiment Analysis Using Hybrid CNN-LSTM Approach," *2021 5th International Conference on Information Systems and Computer Networks*, 1-6, (2021).

Y. Wang, J. Zhu, Z. Wang, F. Bai, J. Gong. Review of applications of natural language processing in text sentiment analysis. *Journal of Computer Applications*, 2022, 42(4): 1011-1020.

Yuan, J., Wu, Y., Lu, X. et al. Recent advances in deep learning based sentiment analysis. *Sci. China Technol. Sci*. 63, 1947–1970 (2020).

Zhao, Jinghua et al. 'Sentimental Prediction Model of Personality Based on CNN-LSTM in a Social Media Environment'. 1 Jan. *Artif Intell Rev*: 3097 – 3106 (2021).