CSE 587 Midterm Project: Semi-Supervised CNN for Movie-genre Classification

KYEONG-HOON LIM

Github Link: https://github.com/kyeongHoony/CSE587

1. Introduction

As we know, RNNs generally perform better than CNNs in natural language processing (NLP) tasks [1]. However, due to the sequential nature of RNNs, they cannot be trained in parallel, which makes the training process longer. In contrast, CNNs can be computed in parallel, making them a viable alternative deep neural network (DNN) model. Moreover, [2] and [3] introduce the capability of CNNs for specific tasks within NLP. Thus, in this project, I focused on using CNNs for text classification.

In CNN model, supervised learning is commonly used. For this learning, the training data should include the label in themselves. However, mapping a label is a laborious task. Therefore, there is much work for unsupervised learning such as K Means-clustering, Hierarchical Clustering, and auto-encoder-like model. However, the challenge of this unsupervised learning occurs during a test. As this model did not receive any label (ground truth), the scoring is performed by comparing the similarity of input and output. Therefore, it is different with a test in supervised learning which is simple matching with the input label and predicted label. Thus, semi-supervised learning is introduced which utilizes two types of data (with label and without label).

Semi-supervised learning can be done in various ways, but pseudo-labeling [4] is a simple yet powerful method. However, since pseudo-labeling accumulates model errors, it can degrade accuracy. To address this issue, I combined two techniques: CNN and FastText [5], a recent word embedding model. By integrating these two techniques, I was able to maintain accuracy even with data containing fewer labels.

2. Problem Definition and Dataset creation

Problem 1: *Supervised learning requires data with a label, which can be problematic. Therefore, to mitigate this I use the pseudo-labeling technique to mitigate this.*

CNN can be useful in some specific tasks of NLP [2], [3]. The common use case of CNN is supervised learning with data having a label. However, mapping a correct label can be problematic in terms of the labor of matching label with data and the size of training data. In order to mitigate this problem, a pseudo-labeling technique is proposed, one of the semi-supervised learning techniques [4], and I used this technique for my project.

Problem 2: *Pseudo-label technique make CNN model trainned with a part of label. Instead, it makes the accuracy of the model lower than baseline.*

Pseudo-labeling has an inherent problem, it can accumulate errors of the model. Because this technique utilized the model to predict the pseudo-label of the given sentence, the model's errors can largely affect accuracy. CNN model using the pseudo-label technique shows lower accuracy than the base CNN model. Thus, I combined the FastText which is the recent word embedding technique to mitigate this effect.

Dataset: For the Dataset I used IMDb training data as training data, and IMDb test data as verification data, for each data there are 25K reviews. For test, I used SST-2 verification, which have 872 reviews.

The reason why I chose this movie review dataset is that from paper, it states that CNN has strength in simple sentence classification. Therefore, I chose movie review which can be

categorized as positive or negative as dataset.

3. Design

For this project, I used a 3-layer CNN model and after each layer I used max-pooling. As an activation function, I used ReLU and at the last layer, I used softmax.

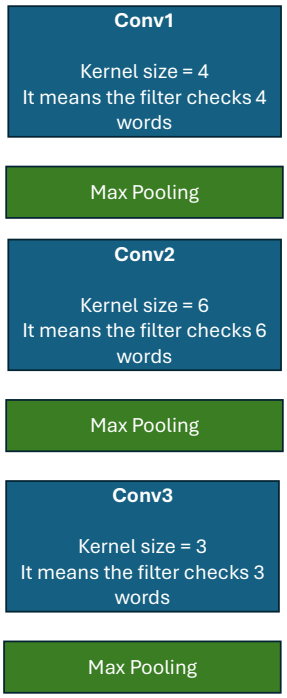

Fig. 1. Overview of CNN model

Initially, I used a lookup-table-based embedding model. First, I used a tokenizer to map each word to an integer. After that, by referencing the table, the code matched each integer to a specific vector.

For the semi-supervised model, I used a prediction-based method. First, I set the ratio of labeled and unlabeled data within the total training dataset (IMDb). Initially, I chose 0.5 as the starting value. Then, I began training using the labeled data. After that, I used the partially trained CNN model to generate label predictions for the unlabeled data. With these predicted labels, I performed additional training on the unlabeled data.

For FastText, I simply replaced the initial embedding layer with FastText. While FastText can be fine-tuned, I did not fine-tune it. This is because fine-tuning did not lead to significant improvement.

4. Evaluation

When we see the effect of FastText, it shows the improvement in accuracy. In my opinion this come from the characteristic of FastText. As we know, FastText uses combined Continuous Bag of Words and N-gram technique. This makes FastText effectively relate the new word with

	CNN Base	CNN + FastText
Accuracy (%)	72	75

	CNN-Semi	CNN-Semi + FastText
Accuracy (%)	68	71

CNN-Semi + FastT Ratio of labeled data	Accuracy (%)	CNN-Semi + FastT Ratio of labeled data	Accuracy (%)
0.1	68	0.6	77
0.2	75	0.7	70
0.3	65	0.8	73
0.4	75	0.9	59
0.5	75		

Fig. 2. Experimental Result

existing word. Moreover, this makes also CNN easily relate novel word to the exact meaning. Therefore, CNN with FastText shows better accuracy than without FastText.

For CNN-Semi with FastText model, I did parameter sweep changing the ratio between labeled data and unlabeled data among the existing training data. The result is surprising, because previously, I think the lower we use labeled data, the lower the accuracy becomes. This is because pseudo-label techniques accumulate error rate. However, the result is not align with this initial thought. I think this comes from the quality of training set. I believe this result makes sense if we assume that higher-quality training data is located toward the front. This suggests that unsupervised learning may be more affected by data quality compared to supervised learning.

Moreover, what is even more surprising is that when the ratio is 0.6, we observe higher accuracy than in supervised learning. This indicates that, as long as data quality is ensured, semi-supervised learning can also achieve significant effectiveness.

This aligns with the fundamental idea of DeepSeek, as the DeepSeek paper also emphasizes the importance of data quality during pre-training. Through this experiment, we were able to confirm its impact.

5. Conclusion

In this project, I combined CNN model, pseudo-labeling, and FastText. As a result, I can maintain the accuracy with partially unlabeled data. For future work, I think embedding layer become more important for CNN model. This is because, CNN model inherently cannot recognize the context of sentences. However, the state of the art embedding technique (ELMo) can recognize the context of the sentence and result in different vector even for same word depending on the context. Therefore, I think this effect of ELMo can compensate weakness of CNN and improve the accuracy of this model.

References

1. W. Yin, K. Kann, M. Yu, and H. Schütze, "Comparative study of cnn and rnn for natural language processing," (2017).
2. Y. Kim, "Convolutional neural networks for sentence classification," (2014).
3. X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," Adv. neural information processing systems **28** (2015).
4. D.-H. Lee *et al.*, "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks," in *Workshop on challenges in representation learning, ICML,* vol. 3 (Atlanta, 2013), p. 896.
5. A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "Bag of tricks for efficient text classification," (2016).