A simple sequence label method for the document-level keyphrase detection

A cute kid

A digital version of the manuscript.

Abstract. The document-level keyphrases detection technology plays an important role in the field of natural language processing. Recent high-performance models have mainly focused on ranking methods. However, such ranking methods are ambiguous and do not make full use of labels. The reason for the ambiguity is that the ranking model can only output phrases with higher scores in sequence, rather than detecting them one by one according to the sequence of keyphrases in the document. In this paper, I propose a GRU-based model with Word2Vec [1] to perform the whole process of document-level keyphrase detection. The whole model can effectively detect the keyphrases in a document sequentially. Experiments on the SemEval 2017 dataset demonstrate that the results of the GRU-based model on different pre-trained variants are different. Further analysis indicates that the GRU-based model suffers from gradient descent instability issues, which pose a significant challenge for previous RNN-based keyword detection methods.

Keywords: GRU-based · Word2Vec · gradient descent.

1 Introduction

The document-level keyphrase detection task refers to identifying the phrases from a given document. Figure 1 demonstrates the whole pprocess of the keyphrases detection. First, send a document to the computer, and then the computer will detect the keyphrases on its own. For the series of approaches, the sequence tag-

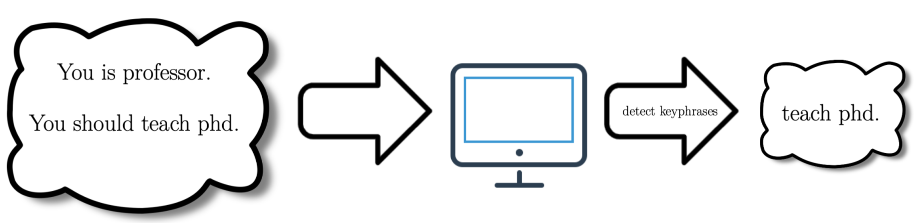

Fig. 1. The keyphrases detection process.

based method is recommended. Since this method can effectively avoid the ambiguity issues caused by ranking models. Therefore, in this paper, I utilize the two strategies of the Word2Vec embedding and combine the GRU-based model to detect the keyphrases. The main contributions of this paper are summarized in three points:

1. Utilize the violin plot to visualize the distribution of the BIO tags.
2. Use SVD to perform dimensionality reduction and visualization analysis on labels.
3. Perform a discrepancy analysis on the model's results in the test set.

2 Methodology

The GRU-based model consists of three parts, the first part is pretraining embedding. The model utilize the Word2Vec embedding to encode the words splited. For the second part, the model uses the GRU and the linear layer to compute the representation of each word. Then, the model send the feats to the next part which is classifier layer to compute. The whole model passes parameters forward using the chain rules and updates parameters backward based on the relationship between feature values and the corresponding labels.

3 Experimental Methodology

This section introduces my experimental settings, including datasets, evaluation metrics, and implementation details.

Dataset. I take one data point from the training set and one from the test set in the SemEval 2017 dataset. And to prevent the model from losing its gradient during training, I divided the document into sentences corresponding to its BIO tags.

Evaluation Metric. I use Accuracy to evaluate the model.

Implemental Details. The Table 1 shows the hypeparameters of the model.

Table 1. model hypeparameters.

hypeparameters	values
torch random seed	66
weight decay	0.0001
learning rate	0.01
epoch	600,1200

4 Results Analysis

Results. Table 2 shows the model with different optimizer and different pre-training embedding accuracy results.

Table 2. model results.

model	Acc.
pure-gru-sgd	$\frac{106}{208}$
CBOW-gru-sgd	$\frac{128}{208}$
skip-gram-gru-sgd	$\frac{86}{208}$
pure-gru-Adam	$\frac{140}{208}$
CBOW-gru-epoch534-Adam	$\frac{126}{208}$
skip-gram-gru-epoch515-Adam	$\frac{145}{208}$

Optimizer analysis. Figue 2 demonstrates the loss curve of the model with different optimizers. The changes in the curve reveal that the GRU-based model training with the Adam optimizer is unstable.

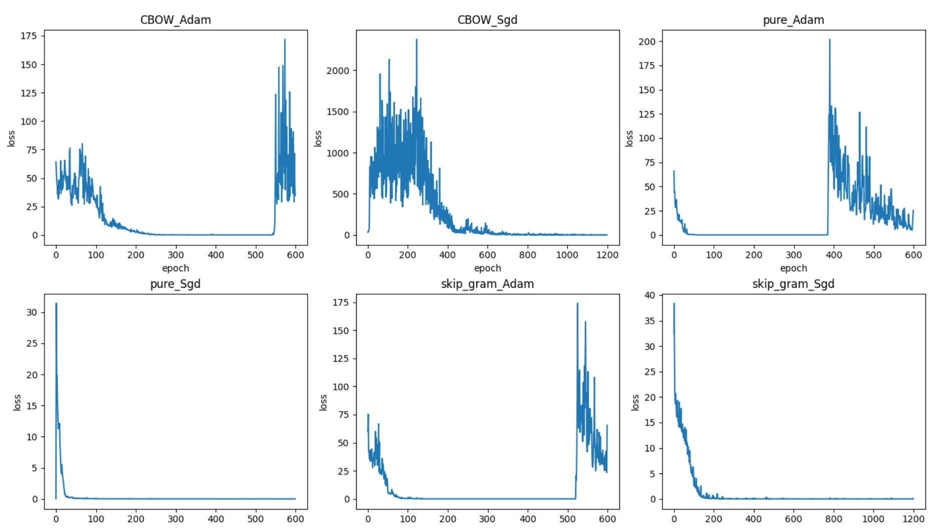

Fig. 2. The optimizer analysis.

Distribution analysis. Figure 3 demonstrates the distribution of the model with its logits. The figure is divided into correctly predicted and incorrectly predicted parts according to the distribution of their corresponding BIO labels. The white horizontal bar and the black bold represent the median and the range of the quartiles corresponding to each distribution.

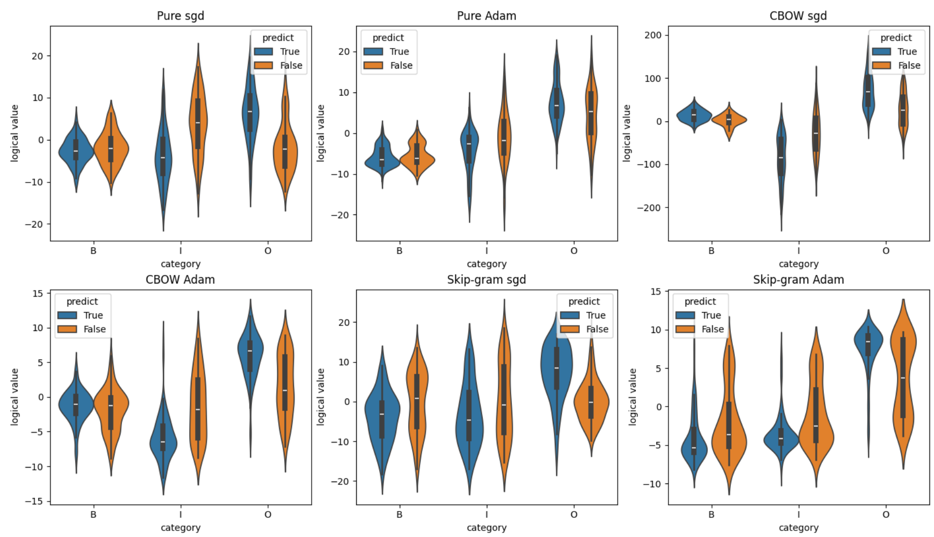

Fig. 3. The distribution analysis.

Dimension reduction analysis. Figure 4 demonstrates the results of reducing the dimensionality of the model's output. Since the difficulty of directly judging

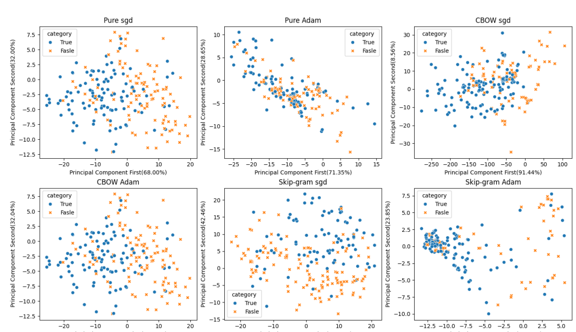

Fig. 4. The dimension reduction analysis.

the relationships between high-dimensional data samples, I used the SVD method to decompose the high-dimensional output of the model, which ultimately allowed the samples to be distributed in a two-dimensional space for easy observation of the relationships.

Discrepancy analysis. Figure 5 demonstrates the results of discrepancy analysis. To perform fold change calculations, I first take the absolute values of the model's predicted values and the corresponding logical values of the true labels. Then, mark points with significant differences via "up" and points with insignificant differences via "down". Mark samples with correct predictions with "no significant". Finally, present the results in the form of a scatter plot.

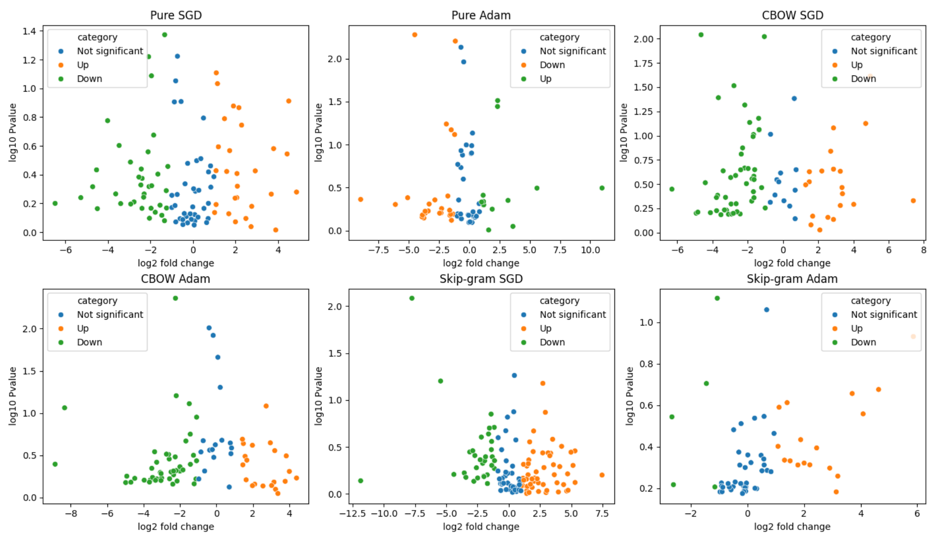

Fig. 5. The discrepancy analysis.

5 Conclusion

This paper propose a GRU-based model for keyphrase detection task and a brief analysis of the results.

References

1. Mikolov, Tomas and Sutskever, Ilya and Chen, Kai and Corrado, Greg S and Dean, Jeff.: Distributed representations of words and phrases and their compositionality. Advances in neural information processing systems, 26 (2013)