Recognizing Arrow Of Time In The Short Stories

Fahimeh Hosseini
Shenakht Pajouh / Sharif University of Technology, Tehran, Iran
fahim.hosseini.77@gmail.com

Hosein Fooladi
Shenakht Pajouh / Sharif University of Technology, Tehran, Iran
fooladi.hosein@gmail.com

Mohammad Reza Samsami
Shenakht Pajouh / Sharif University of Technology, Tehran, Iran
mohammadrezasamsami76@gmail.com

Abstract

Recognizing the arrow of time in the context of paragraphs in short stories is a challenging task. i.e., given only two paragraphs (excerpted from a random position in a short story), determining which comes first and which comes next is a difficult task even for humans. In this paper, we have collected and curated a novel dataset for tackling this challenging task. We have shown that a pre-trained BERT architecture achieves reasonable accuracy on the task, and outperforms RNN-based architectures.

1 Introduction

Recurrent neural networks (RNNs) and architectures based on RNNs like LSTM (Hochreiter and Schmidhuber, 1997) have been used to process sequential data for more than a decade. Recently, alternative architectures such as convolutional networks (Dauphin et al., 2017; Gehring et al., 2017) and attention-based Transformer architecture (Vaswani et al., 2017) have been used extensively and achieved the state of the art result in a diverse range of natural language processing (NLP) tasks. Specifically, pre-trained models such as the OpenAI transformer (Radford et al., 2018; Radford et al., 2019) and BERT (Devlin et al., 2018) which are based on the transformer architecture, have significantly improved accuracy on different benchmarks.

In this paper, we introduce a new dataset which we call *ParagraphOrdering*, and test the ability of the mentioned models on this newly introduced dataset. We have got inspiration from ”Learning and Using the Arrow of Time” paper (Wei et al., 2018) for defining our task. They sought to understand the arrow of time in videos; Given ordered frames from the video, whether the video is playing backward or forward. They hypothesized that the deep learning algorithm should have a good grasp of the physics principles (e.g., water flows downward) to be able to predict the order of the frames in time.

Getting inspiration from this work, we have defined a similar task in the domain of NLP. Given two paragraphs, whether the second paragraph comes really after the first one or the order has been reversed. It is a way of learning the arrow of times in the stories and can be very beneficial in neural story generation tasks. Moreover, this is a self-supervised task, which means the labels come from the text itself.

2 Paragraph Ordering Dataset

We have prepared a dataset, *ParagraphOrdreing*, which consists of around 300,000 paragraph pairs. We collected our data from Project Gutenberg. We have written an API for gathering and pre-processing in order to have the appropriate format for the defined task.[1] Each example contains two paragraphs and a label which determines whether the second paragraph comes really after the first paragraph (true order with label 1) or the order has been reversed (Table 1). The detailed statistics of the data can be found in Table 2.

[1]API for downloading the dataset: https://github.com/ShenakhtPajouh/transposition-data. The implementation of different algorithms: https://github.com/ShenakhtPajouh/transposition-simple

Label	1
First Paragraph	Now they were walking through the trees, one of them carrying him in its huge arms, quite gently. He was scarcely conscious of his surroundings. It was becoming more and more difficult to breathe.
Second Paragraph	Then he felt himself laid down on something soft and dry. The water was not falling on him now. He opened his eyes.

Table 1: A single example of *ParagraphOrdering* dataset.

#Train Samples	294265
#Test Samples	32697
Unique Paragraphs	239803
Average Number of Tokens	160.39
Average Number of Sentences	9.31

Table 2: Statistics of *ParagraphOrdering* dataset.

3 Approach

Different approaches have been used to solve this task, using LSTM, Gated CNN, and BERT. The best result belongs to classifying order of paragraphs using the pre-trained BERT model. It achieves around 84% accuracy on test set which outperforms other models by far.

3.1 Encoding with LSTM and Gated CNN

In this method, paragraphs are encoded separately, and the concatenation of the resulted encodings is going through the classifier. First, each paragraph is encoded with LSTM. The hidden state at the end of each sentence is extracted, and the resulting matrix is going through gated CNN (Dauphin et al., 2017) for the extraction of a single encoding for each paragraph. The accuracy is barely above 50%, which suggests that this method is not very promising.

3.2 Fine-tuning BERT

We have used a pre-trained BERT in two different ways. First, as a feature extractor without fine-tuning, and second, by fine-tuning the weights during training. The classification is completely based on the BERT paper, i.e., we represent the first and second paragraph as a single packed sequence, with the first paragraph using the A embedding and the second paragraph using the B embedding. In the case of feature extraction, the network weights freeze and CLS token is fed to the classifier. In the case of fine-tuning, we have used different numbers for maximum sequence length to test the capability of BERT in this task. We started with using just the last sentence of the first paragraph and the first sentence of the second paragraph for the classification. We wanted to know whether two sentences are enough for ordering classification or not. After that, we increased the number of tokens and accuracy respectively increases. We found this method very promising and the accuracy significantly increases compared to

Model	**Accuracy** (± 0.01)
LSTM+Feed-Forward	0.518
LSTM+Gated CNN+Feed-Forward	0.524
BERT Features(512 tokens)+Feed-Forward	0.639
BERT Classifier(30 tokens / 15 tokens from each paragraph)	0.681
BERT Classifier(128 tokens / 64 tokens from each paragraph)	0.717
BERT Classifier(256 tokens / 128 tokens from each paragraph)	0.843

Table 3: Accuracy on Test set.

previous methods (Table 3). This result reveals that fine-tuning a pre-trained BERT can approximately learn the order of the paragraphs, and thus, the arrow of time in the stories.

References

Alec. Radford, Karthik. Narasimhan, Tim. Salimans, and Ilya. Sutskever. 2018. Improving language understanding by generative pre-training. *Technical report, OpenAI, URL* `https://blog.openai.com/language-unsupervised/`.

Alec. Radford, Jeffrey. Wu, Rewon. Child, David. Luan, Dario. Amodei, and Ilya. Sutskever. 2019. Language models are unsupervised multitask learners. *Technical report, OpenAI*.

Ashish. Vaswani, Noam. Shazeer, Niki. Parmar, Jakob. Uszkoreit, Llion. Jones, Aidan N. Gomez, Łukasz. Kaiser, and Illia. Polosukhin. 2017. Attention is all you need. *Advances in Neural Information Processing Systems*, 5998–6008.

Donglai. Wei, Joseph J. Lim, Andrew. Zisserman, and William T. Freeman. 2018. Learning and Using the Arrow of Time. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Jacob. Devlin, Ming-Wei. Chang, Kenton. Lee, and Kristina. Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Jonas. Gehring, Michael. Auli, David. Grangier, Denis. Yarats, and Yann N. Dauphin. 2017. Convolutional sequence to sequence learning. *arXiv preprint arXiv:1705.03122*.

Sepp. Hochreiter and Jürgen. Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Yann N. Dauphin, Angela. Fan, Michael. Auli, and David. Grangier. 2017. Language modeling with gated convolutional networks. *Proceedings of the 34th International Conference on Machine Learning*, 70:933–941.