NLU Task 2: Story Cloze Task — Report

Lukáš Jendele[*]
ETH Zurich
jendelel@ethz.ch

Ondrej Skopek[*]
ETH Zurich
oskopek@ethz.ch

Vasily Vitchevsky[*]
ETH Zurich
vasilyv@ethz.ch

Micheal Wiegner[*]
ETH Zurich
wiegnerm@ethz.ch

1 Introduction

In the past few years, deep learning models have shown great potential in many areas regarding natural language understanding such as sentiment analysis, language modeling, or machine translation. Unfortunately, research on reading comprehension and logical induction is still in its very early stages. In our work, we attempt to tackle the Story Cloze Task [7]. In this task, one tries to automatically choose the correct ending for a short story. Unlike in other tasks, the Story Cloze task's training and validation phases are conceptually different: during training, the model is given a story consisting of 5 short sentences, whereas for validation, the model is given the first four sentences of a story and two possibilities for the last sentence. The model has to then choose the correct one, i. e. the most logical ending for the story out of the two possible endings. The idea behind this task is to see how well the model can make use of the semantic links between the sentences of the short story and the ending which are necessary to choose the correct answer.

The most challenging part is the lack of negative endings during the training phase. Models trained on the validation dataset [8, 9] have shown much higher accuracy on test data than models trained exclusively on the actual training dataset [8, 11], although the training data is much larger. This problem is caused by the lack of a natural loss function choice for the two different phases. Researchers have been solving this issue either by generating negative samples (using generative methods: GANs [11] or language modeling [8]), or by sampling negative endings from the training dataset (discriminative methods [8]). However, all methods achieve significantly worse results than when training on the validation data.

Another big difficulty is the large natural variability in plausible and implausible endings for a given story. Interestingly, humans are able to achieve an accuracy of 100% for this task, indicating that the task is perfectly solvable despite all the pointed out difficulties.

In our project, we decided to sample the negative endings from the other endings in the training dataset. We reproduced the work by Roemmele et al. [8] and managed to increase the accuracy. The improved accuracy is achieved using multiple approaches.

2 Methodology & Model

Our method follows closely the method described in Roemmele et al. [8]. We use a discriminative binary classifier conditioned on the context sentence. Given the first 4 sentences, the classifier assigns a probability of being a plausible ending to the last sentence. To overcome the lack of negative endings during training, we sample them randomly from other endings the training dataset. Our models utilize the BookCorpus [12] dataset by embedding the story and ending sentences using a pre-trained embedding model. Specifically, we use Skip-Thought sentence embeddings [5].[2] We concatenate the sentence embeddings from both uni-skip and bi-skip Skip-Thought models. Our experiments have shown that using both embedding models yields the best results (as already noted

[*] All authors contributed equally.

[2] We use the implementation in TensorFlow Models by Chris Shallue: `https://github.com/tensorflow/models/tree/master/research/skip_thoughts`

in the original paper). The binary classifier takes the Skip-Thought embeddings, and predicts the last sentence's plausibility (probability of it being a plausible ending). During validation, the model chooses the most probable ending of the two candidates.

2.1 Re-implementation of RNN GRU

We ran a static one-directional 1000-dimensional GRU RNN on the five Skip-Thought embeddings. We use the final state as input to an dense output layer with one output neuron, with a sigmoid activation function – which models the conditional probability that the last sentence is plausible.

After reimplementing the original work, we investigated the impact of replacing the original GRU cell with more powerful LSTM cells and a basic RNN cell (dense layer and $\tanh$ activation). Since Dropout [10] has been shown to improve performance of models in many use cases, we tried to apply a Dropout layer on the inputs of the GRU cell, but the effects on performance were not significant.

2.2 RNN shifted negative endings

Inspired by a forum post[3] claiming that misspellings in word embeddings can be calculated using the difference between a misspelled word and a correct word, we tried to generate sentence embeddings of the negative endings for the positive endings in the training dataset.

Using the first 100 stories in the evaluation set, we calculated the average difference between the wrong and correct endings. We then trained our binary classifier with negative endings calculated as the sum of correct ending and the pre-calculated average difference. We realize that in high-dimensional spaces, it is more common to use cosine similarity. Unfortunately, we did not come up with an idea of how to sample negative endings with a given cosine distance from the correct ending.

2.3 RNN with attention

A natural step is adding an attention mechanism to the model – both multiplicative (Luong [6]) and additive (Bahdanau [1]). Given that the sequences are short, the performance increase should not be large. The mechanisms were designed for translation in a sequence to sequence way. We use a 5-step sequence on the "encoder" side and a 1-step sequence on the "decoder" side (the final RNN state). We compute alignments and successively the averaged attention state, and concatenate it to the RNN's final state before passing both to the fully-connected output layer as previously mentioned.

2.4 Temporal CNN

Instead of an RNN, we ran several temporal convolutions with different kernel sizes (3,4,5) on these embeddings in parallel, similar to Kim [4]. Lastly, we concatenate the CNN feature maps and use a 300-output hidden dense layer with Dropout. The motivation for Temporal CNN was a potential speed-up due to parallelization that is not possible with Gated Recurrent Units (GRUs), at the expense of a higher number of model parameters (30 million in the TCNN model compared to 17.4 million in the GRU model). Due to the number of parameters and the short length of sequences (5), the potential parallelization advantage was not fully leveraged, and the TCNN was actually slower during training.

3 Training

All models are trained with a dense layer with one sigmoid-activated output at the end, and a cross-entropy loss function with labels 0 and 1. The label 1 means that the ending is plausible, and 0 means it is not. We sampled random negative endings from other stories with the same ratio 1:6 as in Roemmele et al. We use mini-batches of size 100 and clip gradients to a maximum L_2 norm of 10. In terms of optimization, we tried Adam, AdaDelta, RMSProp, and standard SGD, all of them with learning rate 0.001. Our experiments confirmed that RMSProp achieves the highest scores for all models except for the Temporal CNN model, where Adam performed better. We ran all experiment on GeForce GTX 1080 Ti on ETH Leonhard cluster. Data processing takes about 15 minutes and training a single epoch about 5 minutes (model times differ minimally).

[3] http://forums.fast.ai/t/nlp-any-libraries-dictionaries-out-there-for-fixing-common-spelling-errors/16411/8

Model	Validation accuracy	Test accuracy
RNN GRU [8] Rand-6 (orig.)	0.645	0.632
RNN GRU [8] Rand-3 + Back-1 + Near-1 + LM-1 (orig.)	0.656	0.672
RNN GRU [8] (ours)	**0.703**	0.675
RNN GRU with constant negative embedding	0.682	0.674
RNN LSTM	0.697	**0.697**
RNN Vanilla ($\tanh$)	0.597	0.568
Bi-RNN GRU	0.684	0.678
RNN GRU with Bahdanau [1] Attention	0.697	0.657
RNN GRU with Luong [6] Attention	0.700	0.675
Temporal CNN	0.689	0.661

Table 1: Accuracy scores achieved by different models.

4 Experiments

We trained all models for 10 epochs (TCNN 30 epochs) and evaluated every 2000 steps. We keep the three best checkpoints of every model, according to validation set accuracy. Finally, we measure the accuracy on the test dataset (ROC 2016), using the best checkpoint for each model (Table 1).

Our implementation of Roemmele's GRU model performs significantly better than the original implementation, according to reported accuracies. We believe this might be because of the superior quality of our pre-trained Skip-Thought embeddings.

Generating sentence embeddings for negative endings from the embedding of the correct ending has proven to have potential, however due to the nature of the method we trained with labels in a 1:1 ratio instead of 1:6, and the model converged very quickly and started overfitting. We believe generating more negative samples or decreasing the model complexity would help.

LSTMs generalized better than GRUs (almost no difference between validation and test), which is probably caused by the extra gates in the LSTM. A large-scale comparison of RNN cells confirms that LSTMs achieve the best performance in NLU tasks [3]. A vanilla RNN is too simple to learn meaningful features from context and does not reach 60% accuracy.

The bi-directional GRU falls behind in accuracy because the right context is likely not that important (backward pass). Srinivasan et al. [9] show that the last sentence is most important for this task, which confirms our claim. Both attention models perform just as well as the models without them, hence they might not be useful with the short sequences – the RNN cells have enough capacity to capture the context. We also tried running models without the RNN (just with attention), but they failed to achieve accuracy above 60%.

TCNN reaches comparable results; however, the envisioned performance boost is not noticeable, because the RNN sequences are probably too short (5 nodes), and the sequential processing is not the bottleneck here. On the other hand, TCNN also has about twice as many parameters as the GRU. Overall, our best model (chosen based on validation accuracy) is RNN GRU (ours) and the best checkpoint can be downloaded from PolyBox[4].

5 Future Work

We did not attempt to use generative methods for this task, because of the discrepancy in goals between training to generate positive endings and later evaluating negative endings [11]. Future work might include designing a generative model conditioned on the labels with negative ending sampling, potentially using reinforcement learning as suggested by Fedus et. al. [2], especially because the generator would not differentiable due to the argmax function for predictions.

We believe that the key to solving the task are generative methods with semi-supervised learning. The lack of negative endings changes this task from a simple binary classification to a conceptually

[4] `https://polybox.ethz.ch/index.php/s/yUr8Iga0OLZ8p6B`

hard problem to model. As mentioned above, training a conditional generative model (possibly a GAN) to sample realistic positive and negative endings might prove to be useful as pre-training, after which the model could be left to improve without labels conditioning.

Also a possibility are various semi-supervised approaches leveraging the well-trained embedding space of the Skip-Thought embeddings, similar to our constant negative embedding experiment. With a smart strategy of negative ending re-sampling based on the currently trained model, it could converge to better results.

6 Conclusion

We re-implemented the model from Roemmele et al. [8] and achieved an even higher accuracy. Furthermore, we explored several variations of the model and analyzed their performance. Our experiments confirm that a GRU RNN in combination with pre-trained Skip-Thought embeddings achieves the best results. Even though our reported results are close to state-of-the-art, current models clearly aren't powerful enough with the currently available data and computation to learn a "world-view" notion, and therefore fall short of human performance.

References

[1] Dzmitry Bahdanau, Jan Chorowski, Dmitriy Serdyuk, Philémon Brakel, and Yoshua Bengio. End-to-End Attention-Based Large Vocabulary Speech Recognition. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016.

[2] William Fedus, Ian Goodfellow, and Andrew M Dai. MaskGAN: Better Text Generation via Filling in the _______. *International Conference on Learning Representations (ICLR)*, 2018.

[3] Rafal Jozefowicz, Wojciech Zaremba, and Ilya Sutskever. An Empirical Exploration of Recurrent Network Architectures.

[4] Yoon Kim. Convolutional Neural Networks for Sentence Classification. *arXiv*, 2014.

[5] Ryan Kiros, Yukun Zhu, Ruslan Salakhutdinov, Richard S. Zemel, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. Skip-Thought Vectors. *arXiv*, 2015.

[6] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. Effective Approaches to Attention-based Neural Machine Translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015.

[7] Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. A Corpus and Evaluation Framework for Deeper Understanding of Commonsense Stories. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*, 2016.

[8] Melissa Roemmele, Sosuke Kobayashi, Naoya Inoue, and Andrew M Gordon. An RNN-based Binary Classifier for the Story Cloze Test. In *Proceedings of the 2nd Workshop on Linking Models of Lexical, Sentential and Discourse-level Semantics*, pages 74–80, 2017.

[9] Siddarth Srinivasan, Richa Arora, and Mark Riedl. A Simple and Effective Approach to the Story Cloze Test. *arXiv*, 2018.

[10] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014.

[11] Bingning Wang, Kang Liu, and Jun Zhao. Conditional Generative Adversarial Networks for Commonsense Machine Comprehension. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, pages 4123–4129, 2017.

[12] Yukun Zhu, Ryan Kiros, Richard Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning Books and Movies: Towards Story-like Visual Explanations by Watching Movies and Reading Books. *arXiv*, 2015.