Story Cloze Test

Lea Fritschi **Beat Hubmann** **Maya Vögeli** **Lukas Wampfler**

1 Introduction

The given task was to design and implement an attempt at the 'Story Cloze Test' [3], a framework for evaluating story understanding and script learning. After some disappointing initial attempts with an attention-augmented encoder-decoder model, we decided to apply the idea to learn sentence vector representations to predict the context of a sentence [1] to the 'Story Cloze Test'.

2 Methodology

In our implementation, we follow the approach of **quick thoughts**, as proposed in [1]. Unlike the approach of generating a target sequence from an input sentence commonly used for this class of problems, our chosen approach attempts to choose the correct target sentence of an input sentence from a set of given candidate sentences based on learned sentence encodings. A massive benefit of this approach is that it allows for a large (i.e. rich) vocabulary size as no target sequence needs to be generated from the encoded input and thus there is no efficiency and numerical stability bottleneck at that point. Also, this approach uses the very simple and efficient classifier $c(u, v) = u^\top v$ to force the encoders to learn relevant sentence representations. Finally, the approach not only allows for but actually relies on large minibatch sizes and thus can be trained efficiently.

3 Model

Our model follows closely what has been described in [1]. Therefore, our model description is taken and adapted from [1]:

Let f and g be two parametrized functions that each take a sentence as input and encode it into a fixed length vector. Let s be a given sentence. Let S_{ctxt} be the set of sentences appearing in the context of s in the training data (in our case: for the second to fourth sentence of a story, the context consists of two sentences: the one sentence before and the one after. For the first and last sentence of each story however, the context consists of one sentence only: the second resp. the fourth one) Let S_{cand} be the set of candidate sentences considered for a given context sentence $s_{ctxt} \in S_{ctxt}$. In other words, S_{cand} contains a valid context sentence s_{ctxt} (ground truth) and many other non-context sentences (in our case, we used the minibatch of 100 sentences as S_{cand}), and is used for the classification objective.

For a given sentence position in the context of s, the probability that a candidate sentence $s_{cand} \in S_{cand}$ is the correct sentence (i.e. really appearing in the context of s) for that position is given by

$$p(s_{cand}\,|s, S_{cand}) = \frac{\exp(c(f(s), g(s_{cand})))}{\sum_{s' \in S_{cand}} \exp(c(f(s), g(s')))}$$

where c is a scoring function/classifier. We use $c(u, v) = u^\top v$ as a classifier.

The training objective maximizes the probability of identifying the correct context sentences for each sentence in the training data D.

$$\sum_{s \in D} \sum_{s_{ctxt} \in S_{ctxt}} \log\, p(s_{ctxt}\,|s, S_{cand})$$

For the encoding functions f and g, we settled on bidirectional RNNs with 600 GRU cells each. When evaluating the trained model, the fourth story sentence is fed to the encoder f and the classifier then decides which of the two candidate endings encoded by encoder g is more probable (see figure).

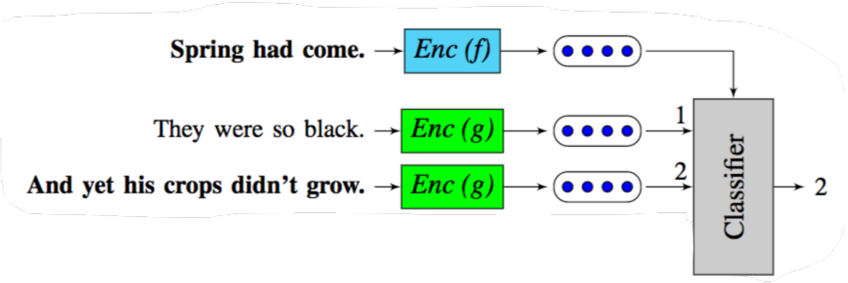

Figure 1: During validation, we use a classifier to decide which of the candidate sentences is the correct ending of the story; figure taken from [1] and adapted

4 Training

For trainable word embeddings, we based on with GloVe pre-trained word embeddings in 300 dimensions [4]. As training data, we exclusively used the given corpus of 88161 five sentence stories. In these stories, there were no alternative (wrong) ending sentences. A context scope of +/- 1 was used, where the previous and the next sentences are predicted given an input sentence. After measuring accuracy on the validation set, we chose hyperparameters to be a minibatch size of 100 sentences (i.e. 20 cloze stories with 5 sentences each) with a dropout rate of 0.25. We trained over 5 epochs to minimize cross-entropy with Adam Optimizer at a learning rate of $1e-3$. Due to the efficient nature of this approach, total training time on a single GTX 1080 Ti GPU was well under an hour.

5 Experiments

On the given validation set, we achieve an accuracy of 0.612 using the parameters mentioned above. The trained model then achieves an accuracy of 0.618 on the assigned test set. This result is comparable with respect to what has been published earlier [2, 5].

6 Conclusion

Due to the limited scope and time of this project, we felt we couldn't fully exploit the benefits of our chosen approach. For further investigation, the use of more sophisticated embeddings (case sensitive; Skip-thoughts) as well as using more sophisticated RNN encoding functions would look promising.

References

[1] L. Logeswaran and H. Lee. An efficient framework for learning sentence representations. *CoRR*, abs/1803.02893, 2018. URL http://arxiv.org/abs/1803.02893.

[2] N. Mostafazadeh, N. Chambers, X. He, D. Parikh, D. Batra, L. Vanderwende, P. Kohli, and J. Allen. A corpus and cloze evaluation for deeper understanding of commonsense stories. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 839–849, San Diego, California, June 2016. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/N16-1098.

[3] N. Mostafazadeh, N. Chambers, X. He, D. Parikh, D. Batra, L. Vanderwende, P. Kohli, and J. F. Allen. A corpus and evaluation framework for deeper understanding of commonsense stories. *CoRR*, abs/1604.01696, 2016. URL http://arxiv.org/abs/1604.01696.

[4] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014. URL `http://www.aclweb.org/anthology/D14-1162`.

[5] M. Roemmele, S. Kobayashi, N. Inoue, and A. M. Gordon. An rnn-based binary classifier for the story cloze test. 2017.