# NLU Project 2: Story Cloze Test

**Akmaral Yessenalina*** ETH Zurich yakmaral@ethz.ch
**Vignesh Ram Somnath*** ETH Zurich vsomnath@ethz.ch
**Ritu Sriram*** ETH Zurich rsriram@ethz.ch
**Meet Vora*** ETH Zurich voram@ethz.ch

## 1 Introduction

Building systems that can understand stories or compose meaningful stories has been a long standing ambition in natural language understanding. Story understanding involves learning meaningful associations and commonsense knowledge from the underlying narrative structure. One such framework that attempts to evaluate story comprehension is that of Story Cloze Test [10]. The task involves associating a 4 sentence story with its correct ending. The training and validation phases in the Story Cloze test are structured differently – the training set consists of the 4 sentence story and the correct ending while the validation set consists of a 4 sentence story with 2 possible choices for the ending. The model has to then choose the correct ending, i.e. the more logical ending out of the two choices. The intent behind the task is to evaluate how well the model learns the semantic associations between the sentences in the story and the endings to choose the correct one.

One of the major challenges in this task is the lack of negative endings in the training set. Attempts have been made to generate negative endings using generative models (GANs [17] or language modeling [13]) or employing different sampling strategies on the training set endings (Roemmele et al. [13]). Despite these attempts, there exists a distributional difference between the (augmented) training and the validation sets, explaining results of previous research ([15],[3]) that achieve higher accuracies by training on the validation set, even though the training set is much larger. Humans, though are able to achieve 100% accuracy on this task, indicating it is perfectly solvable despite the above issues.

In this work, we sample negative endings from other endings in the training set. We investigate different associative strategies between the story context and the endings and replicate models presented in Roemmele et al [13]. and Srinivasan et al [15]. We also explore various extensions to their models, thus managing to increase the accuracy.

## 2 Methodology

Our work largely follows the methods described in Roemmele et. al [13], but we also investigate a different association strategy, as mentioned in Srinivasan et. al [15]. Both strategies build discriminative classifiers that use the context and ending to output a probability of how plausible the ending is. The methods based on Roemmele et al. [13] encode the story and ending together into a single vector used for classification, while the methods based on Srinivasan et al encode the story separately and then learn an association with the ending for classification. Unlike Srinivasan et al. [15], where the authors use the validation set for training, our work uses the training set. To compensate for the lack of negative endings during training, we sample them randomly from other endings in the training set. However, purely for the purpose of comparison, we also train all models on the validation set. (See Table 2.)

Our models utilize the BookCorpus dataset [19] for embedding the story and endings using a pretrained model. In particular, we use the SkipThoughts [8][2] sentence embeddings with a focus on

*All authors contributed equally.

[2]We use the SkipThought embeddings from `https://github.com/tensorflow/models/tree/master/research/skip_thoughts`

the concatenated embeddings from uni-skip and bi-skip SkipThought models, shown to have achieved better results than using either embedding. We refer to these embeddings as SkipThoughts-both or STB henceforth. We also compare the performance of SkipThoughts to the Universal Sentence Encoder embeddings from Cer et al. [2] [3]. These embeddings are referred to as USE henceforth.

During inference time, we make a forward pass with the story and each of the two possible endings. For models based on Roemmele et al. [13], the correct ending is chosen as the one with higher probability. For those based on Srinivasan et al. [15], we look at the probability of being right for both endings, and choose the ending with the higher probability. For models trained on the training set, we report the accuracies on the validation and the Story Cloze test set. For the ones trained on the validation set, only the accuracies on the Story Cloze test set are reported. Models trained on the validation set are not considered for final predictions. We use an ensemble of 3 models with the highest validation accuracy, and adopt a majority voting scheme, for submitting our final predictions.

# 3 Model

## 3.1 RNN Binary Classifier (based on Roemmele et al. [13])

### 3.1.1 GRU-RNN and Variants

We implement the uni-directional, static 1000 dimensional GRU RNN that takes as inputs the embeddings of the story and its ending. The final state from the GRU is then used as input to a Dense layer with one hidden neuron and a sigmoid activation. The output from the Dense layer is the probability that the given ending is plausible conditioned on the story context.

After reimplementing the original model, we also investigated the impact of replacing the original GRU cell with more powerful LSTM cells and also the Vanilla RNN cell.

### 3.1.2 BiDirectional GRU-RNN and LSTM-RNN

A natural extension of the current setting would be to incorporate bidirectionality through forward and backward RNN cells. This allows the model to learn associations not only between the story and ending, but also between the ending and the story, thus in theory allowing for better predictions. We use the concatenation of the forward and backward hidden states, and feed it as input to the Dense layer as described in Section 3.1.1.

### 3.1.3 Incorporating Attention

We further extend our recurrent models and implement attention mechanisms to allow for a better representation of the story. We treat the RNN hidden states of the story as the encoder hidden states, and the ending hidden state as the decoder state. In Seq2Seq terms, this would correspond to a 4-step encoder and single-step decoder. We use both the additive (Bahdanau [1]) and multiplicative (Luong [9]) attention mechanisms. and feed the concatenation of final hidden state and attention state, as the input to the final-dense layer.

## 3.2 Feed Forward Classifier (based on Srinivasan et al. [15])

The authors use story context in three ways - Last Sentence, Full Context and No-Context. We use only the last sentence and full context methods and ignore the no context mode, as the results were not better. After associating the story context and the ending, the resulting encoding is used as input to a multi layered feed-forward network with two neurons and a softmax activation as the output. The two outputs indicate the probabilities of whether the ending is plausible or not, and add to one.

In the Last Sentence based context, the embedding of the last sentence in the story is added to the embedding of the ending, which becomes the input for the feedforward network. In the full context mode, the story sentences serve as inputs for a static uni-directional GRU RNN, with the dimension equal to the embedding size. The final hidden state from the GRU is then added to the ending embedding and used as input for the feedforward network. We also experimented with using more powerful LSTM cells instead of the GRU, and did not experiment with attention for these models.

[3] The pretrained model is available in TensorFlow Hub

## 4 Training

All models were trained to minimize the cross entropy loss with labels 0 and 1. The label 1 indicates that the ending is plausible, while 0 means it is not. Models based on Roemmele et al. use a sigmoid activation, single output neuron, while those based on Srinivasan et al. use a softmax activation over 2 output neurons. Accordingly, the labels were converted into one-hot encodings before computing the loss. Negative endings were used in the same ratio of 6:1 as Roemmele et al. We used a mini-batch size of 100 for Roemmele et al. based models, and clip gradients to a maximum $L_2$ norm of 10, and a mini-batch size of 64 for Srinivasan et al. based models, with a maximum $L_2$ norm for gradients of 5. For optimization, we use the RMSProp [6] optimizer with a learning rate of 0.001, which performed better than our other choices of Adam [7] and AdaDelta [18]. All models were trained using both Universal Sentence Encoder and SkipThoughts embeddings, and were run on GeForce GTX 1080 Ti on the ETH Leonhard cluster. Training a single epoch takes about 3-4 minutes for the Roemmele models, and about 15 minutes for the Srinivasan models. We trained all models for 20 epochs and evaluated every 1000 steps. The checkpoints corresponding to the best validation set accuracy were saved for each model.

## 5 Experiments

Our implementation of the GRU-RNN performs significantly better than the original implementation, from reported accuracies. The original implementation uses the SkipThoughts embeddings from `https://github.com/ryankiros/skip-thoughts`, but we use the ones from Google-Research, which are computed using the same ideas. We believe these are of superior quality thus resulting in higher accuracies. Additionally, Universal Sentence Encoder based embeddings significantly underperform compared to their SkipThoughts models. We think this is because of the increased representational capacity of the SkipThoughts models, along with the training paradigm in Cer et al. [2] favoring better overall transfer task performance, rather than better single task performances.

On average, there is not a major improvement in performance using bidirectional models over unidirectional models, with the marginal improvements attributed to increased representational capacity. For both unidirectional and bi-directional GRUs, we note that additive attention generalizes much better than the multiplicative attention or the no-attention variant. A possible reason for this is discussed in Vaswani et al. [16], where they hypothesize that for larger dimensions, multiplicative attention leads to larger dot products that move the softmax into regions of low gradients. In case of additive attention, this dot product explosion is controlled by the tanh function. This also explains why LSTMs with additive attention fail to generalize as well as GRUs as their gradient flows are more restrictive. The feedforward network using the last sentence as context performs the best, in agreement with the results in Srinivasan et, al. [15]. One reason for this could be that the last sentence constrains the space of possible endings. Another reason could be the inherent bias in the creation of Story Cloze stories, which is discussed in Sharma et al. [14].

## 6 Conclusion and Future Work

We re-implemented the model from Roemmele et al. [13] and achieve higher accuracies than those reported. Additionally, we look at different association strategy, based on Srinivasan et al. [15] and reimplement their models. Various extensions to these models are also considered. The training and evaluation for all models are carried out using two different embeddings - SkipThoughts and Universal Sentence Encoder. Overall, we find that the Last Sentence based Feed Forward model (Section 3.2) performs the best, followed by the BiGRU-RNN with no attention and the BiGRU-RNN with additive attention, with all three models using SkipThoughts embeddings.

Future work would include a further comparison with a current state-of-the-art pretrained models like GPT-2 [12], BERT [5] and ELMo [11]. One major reason for the high human performance is the inherent "world-view" notion and commonsense knowledge, which is refined with time and experiences. Another potential direction to consider would be the incorporation of commonsense knowledge base and sentiment analysis units [4] into the neural architecture. This would fall under the paradigm of semi-supervised learning, using the large training set for sentiment prediction pretraining and finetuning on the smaller validation dataset.

| Model | Attention Type | STB | | USE | |
|---|---|---|---|---|---|
| | | Validation Accuracy | Test Accuracy | Validation Accuracy | Test Accuracy |
| RNN GRU | - | 0.692 | 0.662 | 0.654 | 0.632 |
| RNN LSTM | - | 0.69 | 0.685 | 0.66 | 0.654 |
| RNN Vanilla | - | 0.6 | 0.558 | 0.578 | 0.542 |
| RNN GRU | Multiplicative | 0.694 | 0.674 | 0.664 | 0.645 |
| RNN LSTM | Multiplicative | 0.685 | 0.672 | 0.651 | 0.639 |
| RNN GRU | Additive | 0.686 | 0.684 | 0.654 | 0.636 |
| RNN LSTM | Additive | 0.683 | 0.67 | 0.654 | 0.643 |
| Bi-RNN GRU | - | **0.701** | 0.663 | 0.668 | 0.66 |
| Bi-RNN LSTM | - | 0.687 | 0.678 | 0.658 | 0.649 |
| Bi-RNN GRU | Multiplicative | 0.696 | 0.668 | 0.648 | 0.63 |
| Bi-RNN LSTM | Multiplicative | 0.674 | 0.646 | 0.655 | 0.647 |
| Bi-RNN GRU | Additive | **0.697** | 0.691 | 0.668 | 0.66 |
| Bi-RNN LSTM | Additive | 0.691 | 0.663 | 0.651 | 0.641 |
| FFN-FC-GRU | - | 0.688 | 0.695 | 0.662 | 0.641 |
| FFN-FC-LSTM | - | 0.680 | 0.677 | 0.66 | 0.648 |
| FFN LS | - | **0.707** | 0.688 | 0.65 | 0.623 |

Table 1: Accuracy scores using the original training set.
(STB: SkipThoughts-Both, USE: Universal Sentence Encoder)

| Model | Attention Type | Test Accuracy | |
|---|---|---|---|
| | | STB | USE |
| RNN GRU | - | 0.775 | 0.753 |
| RNN LSTM | - | 0.746 | 0.752 |
| RNN GRU | Multiplicative | 0.764 | 0.747 |
| RNN LSTM | Multiplicative | 0.739 | 0.75 |
| RNN GRU | Additive | **0.78** | 0.752 |
| RNN LSTM | Additive | 0.749 | 0.759 |
| Bi-RNN GRU | - | 0.74 | 0.749 |
| Bi-RNN LSTM | - | 0.738 | 0.761 |
| Bi-RNN GRU | Multiplicative | 0.764 | 0.747 |
| Bi-RNN LSTM | Multiplicative | 0.743 | 0.752 |
| Bi-RNN GRU | Additive | 0.74 | 0.726 |
| Bi-RNN LSTM | Additive | 0.741 | 0.708 |
| FFN-FC-GRU | - | **0.778** | 0.755 |
| FFN-FC-LSTM | - | 0.742 | 0.765 |
| FFN LS | - | **0.797** | 0.735 |

Table 2: Accuracy scores by training on the validation set.
(STB: SkipThoughts-Both, USE: Universal Sentence Encoder)

## 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] Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, and Chris Tar. Universal Sentence Encoder. *arXiv e-prints*, page arXiv:1803.11175, Mar 2018.

[3] Snigdha Chaturvedi, Haoruo Peng, and Dan Roth. Story comprehension for predicting what happens next. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1603–1614, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

[4] Jiaao Chen, Jianshu Chen, and Zhou Yu. Incorporating Structured Commonsense Knowledge in Story Completion. *arXiv e-prints*, page arXiv:1811.00625, Nov 2018.

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv e-prints*, page arXiv:1810.04805, Oct 2018.

[6] Geoffrey Hinton, Nitish Srivastava, and Kevin Swersky. Neural Networks for Machine Learning: Overview of mini-batch gradient descent.

[7] Diederik P. Kingma and Jimmy Lei Ba. Adam: A Method for Stochastic Optimization. *International Conference on Learning Representations (ICRL)*, 2015.

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

[9] 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.

[10] 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.

[11] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. *arXiv e-prints*, page arXiv:1802.05365, Feb 2018.

[12] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

[13] 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.

[14] Rishi Sharma, James Allen, Omid Bakhshandeh, and Nasrin Mostafazadeh. Tackling the story ending biases in the story cloze test. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 752–757, Melbourne, Australia, July 2018. Association for Computational Linguistics.

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

[16] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention Is All You Need. *arXiv e-prints*, page arXiv:1706.03762, Jun 2017.

[17] 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.

[18] Matthew D. Zeiler. ADADELTA: An Adaptive Learning Rate Method. *arXiv*, 2012.

[19] 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.