Multi Layer Perceptron for Story Cloze Test

Aneesh Dahiya **Kaan Oktay**

Abstract

In this report we aim to train a Multi Layer Perceptron network on a portion of validation data set of Story Cloze Test and test its performance on test data set. We use sentence embeddings which capture sentence semantics and syntactic properties to train the network. The Model gave an accuracy of around 72% on test set which is over the pure chance accuracy of 50% [3].

1 Introduction

Story Cloze Test (SCT) [3] consists of four sentence long stories and two alternatives for the endings. One ending is correct and matches with the flow and sentiment of the four sentences and the other ending does not. All the story prompts are manually generated. Humans have an accuracy of 100% in telling correct ending apart from the incorrect endings. The task at hand is to design a language model which can differentiate between the two endings at least better than pure chance accuracy of 50%. The best baseline accuracy as given by [3] is 58.5% on test set with deep structured semantic model [1]. In this report we aim to beat this baseline accuracy by training on features extracted from most of the validation set, validating the model using the other proportion of the validation set and testing the accuracy on test set. For our model, we used a simple Multi Layer Perceptron (MLP) network. The features used for training the MLP network are sentence embeddings in skip-thought vector space [2]. In this representation, semantic and syntactic properties of sentences are mapped into a 4800 dimensional embedding space. Therefore, sentences from the same story will share some similarity in this vector space and a simple MLP should be able to capture this similarity.

Skip-thought vector in a nutshell is an encoder-decoder model where the model tries to reconstruct the surrounding sentences of an encoded passage. After the training of this model, decoder is dropped and the the encoder is used to generate sentence embeddings. For analysis in this report, we generated the vector embedding of sentences from already trained encoder on BookCorpus dataset [5] from [2]. A similar approach has been used by [4] where the authors used the embeddings of sentences to train the network. In our approach, we are giving different attentions to the contribution of each sentence embedding in deciding the correctness of ending and these attention weights are trainable. Also when taking full story prompt into consideration for training network, instead of training network on the embedding of whole story at once, each sentence is treated separately and weighted contributions of sentence embeddings are concatenated to a new vector which is fed as input to train the network. This way sentences closer to ending are given weights which increase the accuracy on validation data set.

2 Methodology

The MLP model used in essence is a discriminative model. In validation set, we have four sentences for the story and two sentences for the ending and it is known that which ending is incorrect. For all stories, vector embeddings for each sentence (X_1, X_2, X_3, X_4, X_5) are created. For each story, there are two different ending embeddings (X_5). Since we have two sentences for ending therefore one story makes two samples while training. The embeddings for endings are added to first four sentence embeddings ($X_1 + X_5$, $X_2 + X_5$, $X_3 + X_5$, $X_4 + X_5$, X_5). The sample where correct ending embedding was added is trained against training output of 1 and the sentence where incorrect ending embedding was added is trained against training output of zero. We use 90% of the validation

data to train our network for 1000 epochs. Model parameters are only saved for those epochs which result in an increase in accuracy of the other 10% of the validation set. We perform five experiments on this model :

1. Using ending alone to predict the correct ending (*5*) .
2. Using ending with last sentence (*54*).
3. Using ending with last two sentences (*543*).
4. Using ending with last three sentences (*5432*).
5. Using ending with all four sentences (*54321*).

Here the numbers in italic represent sentences taken into consideration for prediction.

3 Model

The first layer of the network works controls attention and give trainable weights to each sentence. This layer is followed by three hidden layers of 2048, 1025, 512 units each with ReLU activation except for the last one layer which uses a sigmoid for the activation which acts as a binary classifier. We used a drop out rate of 0.2, 0.15 and 0.10 respectively to prevent over fitting.

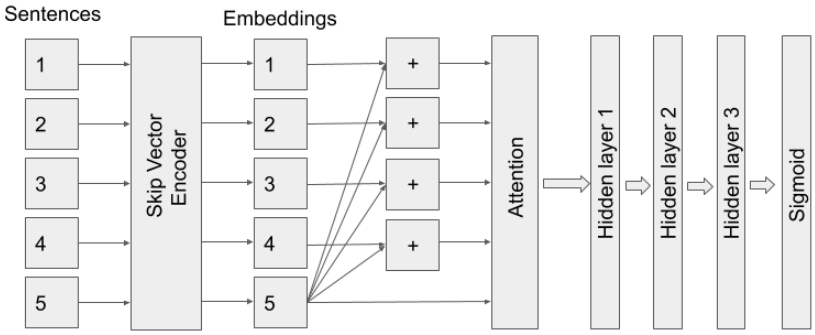

Figure 1: Model Architecture
Sentences are encoded with the encoder to 4800 dimensional vector embeddings, then the embeddings for the ending (5^{th} sentence) are added to each sentence embeddings. Attention layer adjusts the contribution of each sum , e.g for *543* experiment , the corresponding weights for 5^{th}, 4^{th} and 3^{rd} sentence will be non zero rest all will be zero. The attention layer returns on vector of 4800 dimension which then goes to hidden layers followed by softmax layer which puts the output in range [0,1]

4 Training

During training, samples with incorrect ending are trained against zero and samples with correct ending are trained against one. Cross entropy is minimized while training by gradient descent optimization at learning rate of 0.01. The weights in attention layer are trained such that L2 loss is minimized bwtween them. The network was trained for 1000 epochs on 90% of the validation dataset and only those updates were saved which it performed better on the remaining ten percent.

Experiment	Validation Accuracy	Test Accuracy
54321	78.08	72.03
5432	79.04	72.33
543	77.02	71.61
54	78.62	71.08
5	75.53	71.03

Table 1: Validation and Test Accuracy, Average of maximum validation accuracy on the portion not used for training for 5 repetitions of the same experiment and corresponding average test accuracy of these models

5 Experiments

For each experiment the results are shown in the Table 1. Each iteration (total 5) of experiment was done by resampling the data from the validation set of (original SCT), to get better estimate of the accuracy. We also calculated the accuracies on test set of original SCT for all the experiments and their respective iterations.

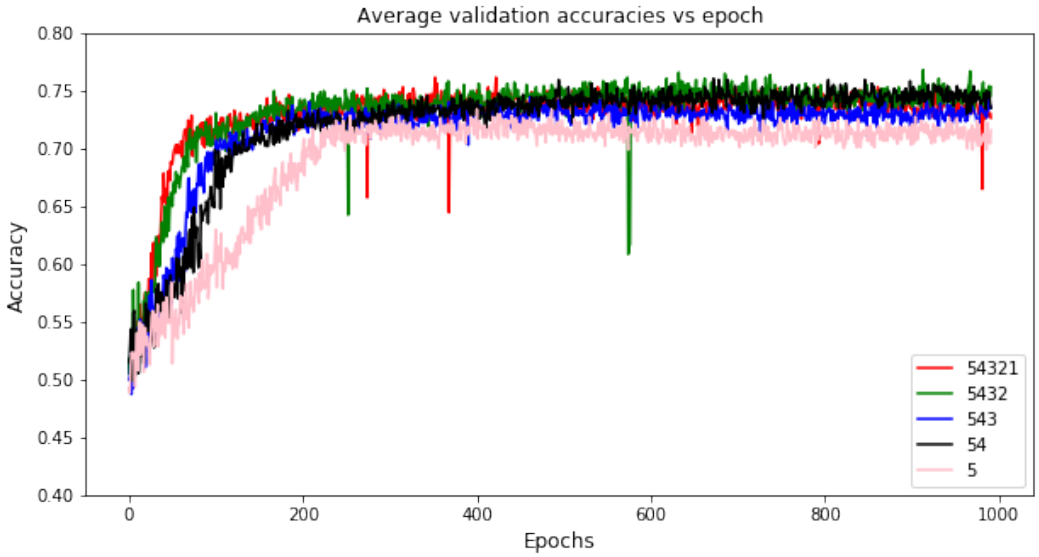

Figure 2: Validation accuracy vs epochs
All the experiments converged somewhere between 70 and 75% accuracy. The sudden dips are due to the fact that for some epochs the prediction for correct and incorrect sentence is equal. It is not counted as correct predictions hence the dips

6 Conclusion

From our experiments, we observe experiment *5432* performed best, but all other experiments gave similar accuracies indicating the model doesn't care much about the attention span or context. Interestingly as pointed out by [4], even with endings alone, model can tell apart the correct ending with probability higher than 0.7. Moreover we can also comment that most incorrect endings share same semantics and sentiment irrespective of other four sentences in a sample.

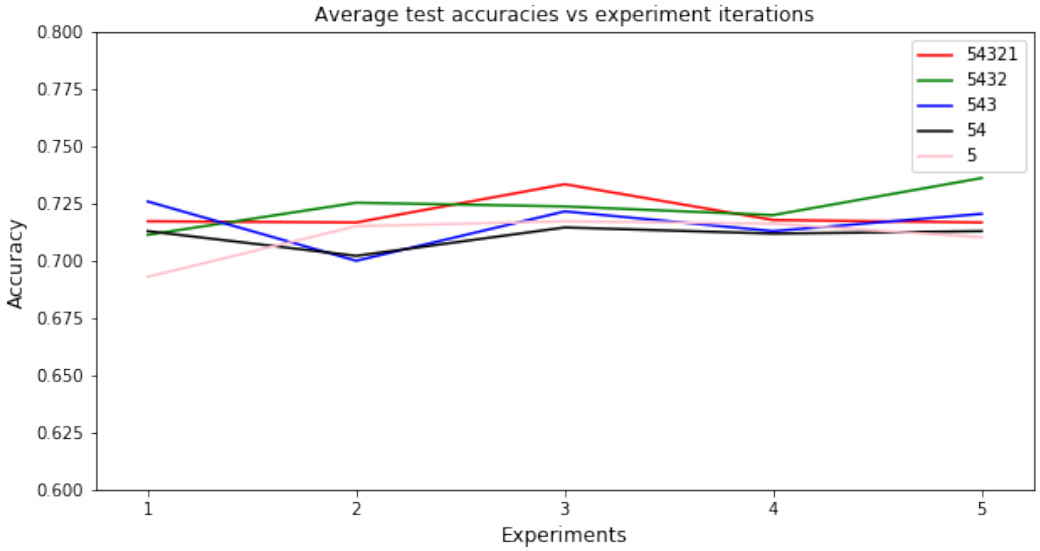

Figure 3: Test Accuracy

References

[1] P.-S. Huang, X. He, J. Gao, L. Deng, A. Acero, and L. Heck. Learning deep structured semantic models for web search using clickthrough data. October 2013.

[2] R. Kiros, Y. Zhu, R. Salakhutdinov, R. S. Zemel, A. Torralba, R. Urtasun, and S. Fidler. Skip-thought vectors. *CoRR*, abs/1506.06726, 2015.

[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.

[4] S. Srinivasan, R. Arora, and M. Riedl. A simple and effective approach to the story cloze test. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 92–96, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[5] Y. Zhu, R. Kiros, R. S. Zemel, R. Salakhutdinov, R. Urtasun, A. Torralba, and S. Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. *CoRR*, abs/1506.06724, 2015.