Data Augmentation to Improve BERT on Story Cloze

Leopold Franz lfranz@ethz.ch
Felix Graule graulef@ethz.ch
Esref Özdemir esrefo@ethz.ch
Rhea Sukthanker srhea@ethz.ch

1 Introduction

Research in natural language understanding (NLU) has seen a rapid progress especially with the "deep learning era" which helped researchers to move from language processing to language understanding in the true sense. However, current research methodologies in NLU, especially the state of the art models for tasks like sentiment analysis and coreference/event resolution unequivocally suffer from lack of commonsense reasoning. Current NLU models still struggle connecting distant information pieces and understanding the semantic coherence between different portions of the given text. The Story Cloze Task [7] is aimed at gauging these exact capabilities in a model.

The Story cloze test aimed at targeting the challenge of understanding causal and correlational relations in text. It was a part of the LSDMSem'17 shared task [1] for story understanding and script learning. This system provided four sentence stories with two possible endings, one of which is the correct one. The main challenge of this task is that the training set (ROCstories corpus) does not contain negative endings(ie. it is five sentence story corpus). This means that training set by itself is not enough to train a regular classifier. On the other hand the validation and test set of the task poses more of a binary classification problem between the right and the wrong endings.

In this project we explored the following approaches to tackle the task. Firstly we aim at understanding the task and getting a notion of why training on the validation set can introduce problems. Secondly we use the state of the art models like BERT [3] (Bidirectional Encoder Representations from Transformers) to get an idea of how it performs. Thirdly we explore data augmentation strategies inspired by methods in [11] to smartly sample negative endings from the training corpus. Finally we measure how BERT performs when fine-tuned on our augmented datasets.

2 Related Work

Data augmentation techniques have been used to generate negative endings for each sample in the training set [1][11]. This augmented dataset is then used as input to a binary classification task. The main augmentation approaches in [11] are randomly choosing a new ending, choosing a new ending that is closest to the sample ending in some vector space and sampling a new ending using a language model. These three ideas provide a foundation for the proposed methods in this report.

Logistic Regression (LR) on surface and stylistic features of the validation dataset has been widely popular for the story cloze task. The best performing LR model [13] was based on language modelling probabilities of the entire story and linguistic features. Other LR models used frame based, sentiment based language modelling probabilities and topical consistency . In addition sentiment features were widely used too[4]. The most interesting conclusion from these methods is that using the stylistic features only from the endings achieves a much better than random accuracy.

Deep learning has transformed the landscape of NLU and this is no different for the Story Cloze Task. Some approaches which were less dependent on handcrafted features used pre-trained word embeddings like GloVE and word2vec [12]. Further, an LSTM was used to learn the composition of the embeddings. Some other RNN based approaches also relied on sentence level SkipThought embeddings [6][11]. However, it is important to note that deep neural networks are mainly powered by the availability of large datasets. Therefore these approaches which are "trained" on the validation dataset potentially have poor generalization capabilities.

3 Methodology and Ideas

In this section we describe our baselines and the main ideas we explored. In all of our methods we aim at solving a binary classification problem using a discriminative classifier. However, in models that use training set we employ generative techniques to choose or generate incorrect endings.

3.1 Data Exploration

We started with data exploration to gain useful insights into potentially useful features. We observe that on average the correct endings tend to be longer that the incorrect ones. Also the polarity of the correct ending tends to be more positive than the wrong ending. It is easy for a discriminative model like logistic regression trained on such features to overfit. This can be seen from the difference between the cross-validation and test-accuracy. Thus a model trained using only the validation set is not robust enough.

3.2 Classical NLP

Our first baseline is a simple logistic regression model trained on ending features of the validation set. The features we use include length of the sentence, word/character n-gram features, polarities from VADER [5], counts of the occurrences of different NER tags in the data and subjectivity/polarity score from TextBlob. In addition we add some features which take the entire story into account. These features are the one-gram overlap count between the ending and the story and the difference in the polarities between the story and the ending. These features aim at describing a level of similarity between the story and the ending. We pick the ending with the higher correctness probability as predicted by LR to be the correct one.

3.3 Stacked Bidirectional LSTM on Sentence Embeddings

We use an LSTM network on sentence embeddings as our second baseline. We generate sentence embeddings for each sentence in the story using SkipThought vectors [6] and then feed these sequence of vectors to LSTM layers. The model consists of three bidirectional LSTM layers stacked on top of each other and a final dense layer with sigmoid activation function. The hidden state sizes of LSTM layers are 256, 128 and 64 (layer before dense). We use dropout at each layer in the network for regularization. The model is trained only on the validation set. Correct ending for a test sample is predicted by constructing two five sentence long stories and getting the correctness probability of both samples. Then, the ending with higher correctness probability is chosen as the correct ending.

3.4 BERT with Sentence-Pair Classification Task

In order to leverage the recent performance gains in NLP research for our task, we use BERT[3]. It is described by the authors as the *first unsupervised, deeply bidirectional system for pre-training NLP* and has gotten a lot of attention within the field due to to its strong performance on a wide variety of downstream NLP tasks. BERT can be both used for specific tasks that where also used for pretraining, like SQuAD [10], or to generate contextual word embeddings similar to ELMo [8].

We found that BERT's Sentence-Pair Classification task best fits our goal and use it to fine-tune a pretrained version of BERT (BERT-base, uncased). During fine-tuning based on this task, the network expects a tokenized sentence pair with $(sent\_A, sent\_B)$ and a 0/1-label describing whether $sent\_A$ is truly followed by $sent\_B$ (label is 1 if sentence pair connected). At inference time, we again hand the network two sentences and get the probabilities of the sentence pair being unrelated, $P_{neg}(end)$ and being connected, $P_{pos}(end)$. Since this Sentence-Pair Classification task does not perfectly match the Story Cloze task, we adjust the network architecture slightly as described in section 4.

Baseline model	Mean CV accuracy	Story Cloze test accuracy
Logistic Regresssion	63,06%	56,68%
Bidirectional LSTM	N/A	67,00%
BERT fine-tuned	N/A	87,5%

Table 1: Comparison of BERT with Baselines

3.5 Data Augmentation

Up until now both of our baselines and BERT relied only on the validation set. Since this may lead to poor generalization, we decided to make use of the much larger training set. To do this, we tried data augmentation techniques to generate negative samples by only using the training set. Our methods were inspired by [1] and [11].

As baseline strategies, we implemented Random and Backward strategies from [11]. We tried making various modifications to the Nearest strategy from [11] and came up with several methods in the end. We shortly describe each of these strategies below. For a given five sentence sample in the training set

1. Random strategy picks a random fifth sentence from training set as the negative ending
2. Backward strategy picks one of the first four sentences of the same story as the negative ending
3. NearestEndSent2Vec strategy picks the ending from the training set that is most similar to the current ending and that is different than the current ending. Similarity is measured as the cosine similarity of Sent2Vec vectors for the sentences.
4. NearestStorySent2Vec strategy picks the ending of the sample whose first four sentences are the most similar to the first four sentences of the current sample. Similarity is measured as the cosine similarity of Sent2Vec vectors corresponding to the concatenation of the first four sentences.
5. NearestStoryUSC strategy replaces Sent2Vec embeddings with the recently proposed Universal Sentence Encoder [2] embeddings.
6. NearestStoryUSCwithNLP strategy extends the above Universal Sentence Encoder strategy by concatenating features obtained with classical NLP techniques on only the ending sentence of the story.

3.6 Generating Negative Endings

One major drawback of data augmentation by selecting the negative endings from the training set is that the chosen ending does not necessarily have a contextual relationship with the first four sentences of the story in consideration. To overcome this problem we try to generate the endings. This approach of generating the endings is similar in its aim to the Language Model approach in [11]. However, we use the conditional generation operation mode of the much more powerful GPT-2 model [9] to generate sample text given the first four sentences of a story. Then, we use the first sentence generated by the model as the negative ending.

4 Final Model and Training

We selected the architecture for our final model based on the achieved test accuracy on the Story Cloze test set. BERT outperformed all other models by a large margin as can be seen in table 1.

We use a slightly adapted version of BERT with an additional classification layer on top. Our architecture is closely related to an existing pipeline used for sentiment analysis on movie reviews [1].

When fine-tuning the pretrained BERT model, we take a naive approach and pass the whole story context (first four sentences) as $sent\_A$, one of the candidate endings as $sent\_B$ and the respective label describing whether the passed ending is the incorrect (0) or correct (1) one. We then optimize by minimizing the cross-entropy loss over the labels. Hence, there is no Story Cloze specific, additional loss or feedback during fine-tuning. We fine-tune for 3 epochs with a learning rate of $2 \cdot 10^{-5}$, batch size of 32 and maximal sequence length of 128.

During inference, we need to fit the Sentence-Pair Classification task to the Story Cloze task. To do so, we made the following adjustments: Again, instead of feeding two single sentences, we pass the whole story context (first four sentences) as $sent\_A$ and one of the candidate endings as $sent\_B$. Thus, for every Story Cloze element (context + two candidate endings), we pass through the network

[1] `https://colab.research.google.com/github/google-research/bert/blob/master/predicting_movie_reviews_with_bert_on_tf_hub.ipynb`

twice resulting in four probabilities: $P_{neg}(end_1), P_{pos}(end_1), P_{neg}(end_2), P_{pos}(end_2)$. We then only compare the positive prediction probabilities in order to find the right candidate ending, so if $P_{pos}(end_1) > P_{pos}(end_2)$ we predict the first ending to be the correct one and vice-versa.

5 Experiments

In order to assess the quality of the different data augmentation methods we developed, we run the BERT fine-tuning procedure described above for each of the augmented datasets separately. The results are shown in table 2. We report three metrics: the accuracy of Story Cloze task where a sample is predicted correctly if the model picks the correct ending, the accuracy of identifying correct and wrong endings where a sample is predicted correctly if the model predicts the correct ending as correct and wrong ending as wrong and the ratio of cases where the positive predictions for the two candidate endings are very close to each other, so whenever $|P_{pos}(end_1) - P_{pos}(end_2)| < 0.01$. If this ratio is high, it indicates that the sampled endings are both reasonable endings to the story, making it very hard for the model to decide which one fits better. This is the case for the Backward and Random Nearest strategies and also for the endings generated with GPT-2.

Exp	Fine-tuned dataset	Story Cloze accuracy	Predict endings accuracy	Narrow Endings
1	No fine-tuning	46,7%	8,2%	4,8%
2	Story Cloze Validation	87,5%	68,0%	11,4%
3	Backward	44,5%	0,0%	44,5%
4	Random	56,1%	7,5%	46,4%
5	NearestEndSent2Vec	55,9%	10,7%	4,2%
6	NearestStorySent2Vec	66,7%	15,4%	6,5%
7	NearestStoryUSC	71,2%	25,8%	4,2%
8	NearestStoryUSCwithNLP	64,9%	13,5%	11,0%
9	Generated ending GPT-2	59,4%	16,7%	36,4%
10	**Final Model**	**80,9%**	**48,8%**	**10,8%**

Table 2: Results when fine-tuning BERT on different type of augmented data.

Experiment 1 suggest that without fine-tuning, the model performs very poorly (below random baseline). Experiment 2 shows that fine-tuning on the Story Cloze validation set performs best with respect to the Story Cloze test set. Experiments 3 and 4 show that choosing an ending from a random other story is better than using a non-ending sentence from the same story. From experiments 5 and 6 one can see that taking the whole story context into account boosts performance. USC vector space is better compared to Sent2Vec with respect to Story Cloze accuracy as can be seen in experiment 7. However, adding classical NLP features to USC lowers the overall performance (experiment 8). Finally, the problem with generating endings (experiment 9) is that while the ending fits into the context, there is no guarantee that it would be a negative ending. Therefore, positive and negative endings are not easily separable. This can be seen from the high ratio of narrow endings. Nevertheless, the network is able to extract some amount of useful information from these endings.

Based on these conclusions, we decided to train our final model on a mixture of the datasets we evaluated (experiment 10). From looking at the results in table 2, we see that all of the available data in the Story Cloze validation should be used. We also use the highest scoring nearest strategy, NearestStoryUSC, and the endings generated using GPT-2. Due to memory constraints, we only use half the NearestStoryUSC and GPT-2 data, while we replicate the Story Cloze validation data such that each data source accounts for a third of the overall data. As expected, the overall performance with respect to the Story Cloze test set is lower than for experiment 2, since we bias the data away from the test set. However, we still use the model trained on mixed data, since we expect it to be more robust to changes in the test set, thus performing better on the data distributed by the organizers.

6 Conclusion

In this project we explored the effectiveness of BERT on Story Cloze task. BERT was able to significantly outperform our baseline models when fine-tuned on the given validation set. Subsequently we tried to increase its generalizability by combining it with data augmentation methods.

References

[1] Michael Bugert, Yevgeniy Puzikov, Andreas Rücklé, Judith Eckle-Kohler, Teresa Martin, Eugenio Martínez-Cámara, Daniil Sorokin, Maxime Peyrard, and Iryna Gurevych. Lsdsem 2017: Exploring data generation methods for the story cloze test. In *Proceedings of the 2nd Workshop on Linking Models of Lexical, Sentential and Discourse-level Semantics*, pages 56–61, 2017.

[2] Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, et al. Universal sentence encoder. *arXiv preprint arXiv:1803.11175*, 2018.

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

[4] Pranav Goel and Anil Kumar Singh. Iit (bhu): System description for lsdsem'17 shared task. In *Proceedings of the 2nd Workshop on Linking Models of Lexical, Sentential and Discourse-level Semantics*, pages 81–86, 2017.

[5] Clayton J Hutto and Eric Gilbert. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Eighth international AAAI conference on weblogs and social media*, 2014.

[6] Ryan Kiros, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Skip-thought vectors. In *Advances in neural information processing systems*, pages 3294–3302, 2015.

[7] Nasrin Mostafazadeh, Michael Roth, Annie Louis, Nathanael Chambers, and James Allen. Lsdsem 2017 shared task: The story cloze test. In *Proceedings of the 2nd Workshop on Linking Models of Lexical, Sentential and Discourse-level Semantics*, pages 46–51, 2017.

[8] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proc. of NAACL*, 2018.

[9] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 1:8, 2019.

[10] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy S. Liang. Squad: 100, 000+ questions for machine comprehension of text. In *EMNLP*, 2016.

[11] Melissa Roemmele, Sosuke Kobayashi, Naoya Inoue, and Andrew 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.

[12] Niko Schenk and Christian Chiarcos. Resource-lean modeling of coherence in commonsense stories. In *Proceedings of the 2nd Workshop on Linking Models of Lexical, Sentential and Discourse-level Semantics*, pages 68–73, 2017.

[13] Roy Schwartz, Maarten Sap, Ioannis Konstas, Leila Zilles, Yejin Choi, and Noah A Smith. Story cloze task: Uw nlp system. In *Proceedings of the 2nd Workshop on Linking Models of Lexical, Sentential and Discourse-level Semantics*, pages 52–55, 2017.