Designing an Automatic Story Evaluation Metric

Stanford CS224N Custom Project

Hannah Huddleston
Department of Computer Science
Stanford University
hannhudd@stanford.edu

Karen Ge
Department of Symbolic Systems
Stanford University
kge@stanford.edu

William Shabecoff
Department of Computer Science
Stanford University
wis23@stanford.edu

Abstract

The rise of natural language models has led to their application to a plethora of complex tasks, including dialog systems and story generation. Such novel tasks necessitate automatic evaluation metrics for the plausibility of such machine-generated stories and thus aid in training these models. However, reference-based evaluation approaches such as BLEU and ROUGE do not perform well here because of the open-ended nature of story generation. Thus this paper proposes a novel framework for such an automatic story metric using emotional states and a commonsense model to measure the coherence and meaning-making ability of a story. Our model had the most success using a BERT-base-uncased with a linear layer on the Story Cloze task.

1 Key Information to include

- TA Mentor: Michihiro Yasunaga
- External Mentor: Lisa Li
- External Collaborators (if you have any): None
- Sharing project: None

2 Introduction

Telling stories is a natural and easy task for humans, but a challenge for natural language models. To generate a good story, a model must not only produce coherent text, but also craft a logical plot that follows a main character through the beginning, middle, and end of the story. The task of training a story generation model becomes increasingly difficult due to the open-ended nature of stories. There are many correct ways to tell a story, so reference-based evaluation metrics such as BLEU and ROUGE cannot simply classify a story as “correct" or “incorrect." Therefore, there exists the need for better automatic story evaluation metrics that take into account both the sentence level and plot level coherence of the story. This is a significant challenge for natural language models because it requires both semantic and commonsense understanding.

3 Related Work

The classic story evaluation challenge is the *Story Cloze Test*, in which the first four sentences of a story is given as context and the model must select the “correct" ending out of two choices. Much of

the previous work on the *Story Cloze Test* has focused on semantic or stylistic similarity between the story and the two endings. Mostafazadeh et al. made the first Story Cloze models in 2016, which incorporate features such as word frequency, N-gram overlap, average Word2Vec embeddings, sentiment, and narrative chains [1]. The best performing model from this baseline study was a deep structured semantic model with two neural networks, and it achieved an accuracy of only 0.585 on the test set, while humans were able to correctly choose the ending 100% of the time [1].

Srinivasan et al. (2018) used skip-thought embeddings trained on the BookCorpus dataset in a 3-layer feed-forward neural network, which reached an accuracy of 76.5% [2]. They tested the model with the full story context, no context at all, and just the last context sentence, which performed the best of the three [2]. This result raises questions about the effectiveness of the story evaluation metric and its ability to generalize to data outside of Story Cloze. If the model performs best with only the preceding sentence to the story ending, it is not capturing information about the entire story plot, which could create a problem for longer or more complicated stories. Furthermore, the no-context model achieved an accuracy of 72.6% just by examining the correct and incorrect story endings, which means that the Story Cloze endings may be biased [2].

Schwartz et al. (2017) used an LSTM recurrent neural network language model (RNNLM) that also takes in stylistic features like sentence length, word n-grams, and character n-grams [3]. This combined model achieved an accuracy of 75.2%. The RNNLM features alone performed at 67.7% accuracy, while the stylistic features were 72.4% [3].

The best-performing Story Cloze model to date achieves an accuracy of 91.8% using a a transferable BERT training framework [4]. Error analysis revealed that the model made mistakes when one ending was about mental state while the other was about a next action [4].

Many of these preliminary studies on the *Story Cloze Test* show that stylistic and semantic features can help a model evaluate the coherence and consistency of a story and ending. While these results are promising, they are limited to the specific format of the five sentence Story Cloze stories and may not generalize to larger or more complex stories with higher-level plot inconsistencies. The most successful model by far uses BERT to encode information about the whole story context.

4 Approach

We hypothesize that the plot of a good story is one with a consistent sequence of events and emotional states. Therefore, we pass information about those two features to our model along with BERT embeddings of the sentences in the story. To encode the sequence of events in a plot, we use COMET Commonsense Transformers, which build knowledge graphs about each sentence in the story. To encode the emotional states of the story, we use a multilabel classifier to predict 8 different emotions for each sentence.

COMET:

We used COMET Commonsense Transformers for Automatic Knowledge Graph Generation. COMET is trained on natural language tuples in $\{s, r, o\}$ form, where s is a phrase subject, r is a relation, and o is a phrase object. We used COMET trained on the semantic knowledge graph ConceptNet which relates words and phrases over a variety of common relations like `Causes, HasProperty, HasPrerequisite` among others. $\{s$="Melody saw sharks",r="Causes",o="excitement" $\}$ and $\{s$="Melody saw sharks",r="Causes",o="panic" $\}$ are both examples of such ConceptNet knowledge in the form of COMET tuples.

COMET generates object o given subject s and relation r. We used COMET to generate plot inferences at each sentence in the story. Since running COMET on our entire dataset is somewhat computationally expensive, we decided to focus on *HasPrerequisite* and *Causes* in order make inferences about the logic flow of the plot. For each sentence, we first used spaCy to extract a basic tuple representation of the sentence which consists of the subject, relation, and object of the sentence. We then query COMET with this tuple over the *HasPrerequisite* and *Causes* relations and take the top 3 results using beam search for each relation leaving us with 6 inferences for the sentence.

Here is a full example of this part of the pipeline:

Source Sentence "It was my final performance in marching band."

- **Has Prerequisite** “rehearse", “musical instrument", “band"
- **Causes** “cheer", “applause", “crowd"

Emotional States:

Though stories can contain a range of emotions and surprise endings, we hypothesize that emotional states can be used to measure the consistency of a story plot. In particular, many of the incorrect endings from the Story Cloze dataset do not align with the emotional states from the preceding sentences in the story. Consider the story example in Table 1. In the first four sentences of the story, the characters experience emotions such as joy, excitement, and satisfaction. The correct ending is consistent with those emotions, but in the incorrect ending, the characters experience disappointment.

Story	Correct Ending	Incorrect Ending
A family was driving home from vacation in Florida. They noticed a space museum and decided to stop. They went inside and checked out all of the exhibits. After that, they watched a movie about space.	The family was glad they spotted the museum.	The family felt they had wasted their time.

Table 1: Story Cloze Example

For every sentence in the story, we use a multilabel classifier to predict the presence of psychologist Dr. Robert Plutchik’s 8 primary emotions: joy, sadness, acceptance, disgust, fear, anger, surprise, and anticipation. We train our emotion classification model on Story Commonsense data, which contains over 9k story sentences hand-labeled with Plutchik’s emotions. We decided to use the Story Commonsense dataset over larger datasets like GoEmotions, which has 58k Reddit posts annotated with 27 emotions, because we expected a model trained on story sentence data to generalize better to our task.

To put it all together, our model architecture is described below.

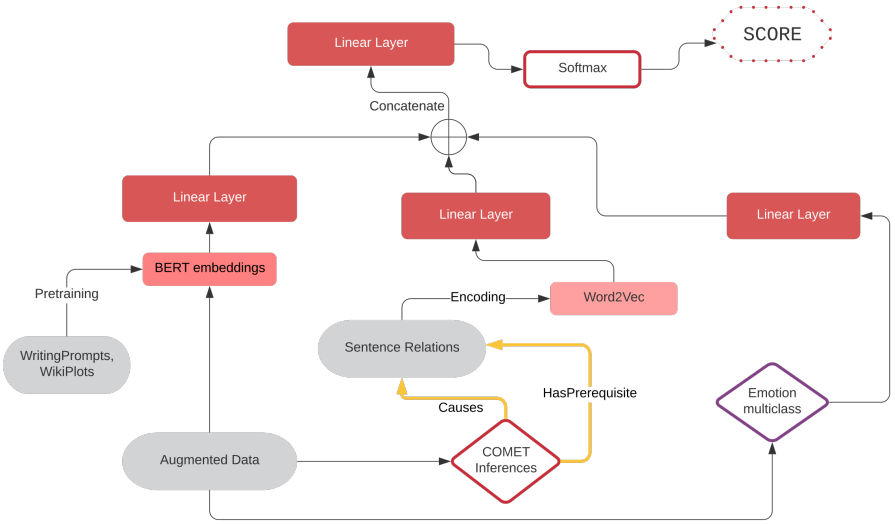

Figure 1: Our Model Architecture

From our augmented and adversarially-generated story examples we encode each sentence s_i as a simple knowledge graph G_i. We then pass the knowledge graph G_i into COMET. From COMET we query a 3-wide beam search for the most likely results from this graph for the relations `Causes` and `HasPrerequisite`. Then we use the GoogleNews word2vec vectors to embed each of these 6 phrases and compute their centroid. For the emotions vectors, we finetuned SqueezeBERT model

along with a linear layer to run a multilabel classifier for the 8 emotions. The hyperparameters of the shape of the linear layers on top of the BERT embeddings of the entire story, word2vec encodings, and emotions vectors was chosen with the experiments described below.

5 Experiments

5.1 Data

The main datasets we used were Story Cloze and ROCStories. ROCStories is a collection of 98,000 5-sentence english commonsense stories. Story Cloze is a smaller dataset of 5-setence stories where each story 3,700 stories where each story has a “good" and “bad" ending sentences. While both the “good" and the “bad" endings have well formed last sentences, the “good" ending should make more sense in the context of the story.

One version of our model was also pretrained on the Writingprompts and Wikiplots datasets. Writingprompts consists of short stories written by users of a story-writing community on Reddit. Wikiplots consists of plot summaries of movies on Wikipedia.Pretraining on this data did not improve performance.

5.1.1 Data Augmentation

Since we had many more positive than negative examples for stories, we augmented our data by creating more adversarial negative examples of stories based on the ROCStories. We had two methods of generating bad stories.

- **Swapped Ending** Swap the last sentence of the ROCstory with the last sentence of some randomly chosen other ROCstory.
  - **Example** “Jenny has a drinking problem. Jenny got arrested for public intoxication. Jenny was in jail for three days. Jenny could not write a check for rent from jail. Finally she got the cord free."
- **GPT-2 Generated Ending** Use the first four sentences of the ROCstory as context and have GPT-2 generate the last sentence.
  - **Example** “David noticed he had put on a lot of weight recently. He examined his habits to try and figure out the reason. He realized he’d been eating too much fast food lately. He stopped going to burger places and started a vegetarian diet. He also began to lose the braids in his head thanks to his behavior."

With these methods we were able to generate adversarial examples of incoherent stories and have a much larger dataset of positive and negative labelled stories.

5.2 Evaluation method

Before training our model we partitioned our data into train, validation, and test data with a 80%, 10%, 10% split. We evaluate on the accuracy of our model in predicting the correct label of a story. We also calculated the F1 score of our models for both the negative (0) and positive (1) labels.

The Story Cloze classification task is also a popular task in the NLP community, so we also considered the results of Mostafazadeh et al. and Li et al. as additional baselines (though we did not surpass the accuracy achieved in Li et al.)[1][4]. It is important to note that other Story Cloze baselines were evaluated on a selection between two story endings, while our model outputs a score to a story and one given ending.

5.3 Experimental details

The multilabel emotions model was finetuned on SqueezeBERT[1]. As shown in Figure 2, we ran 6 sweeps to tune hyperparameters and optimized according to the AUC score. AUC is the area under the ROC curve, which plots the false positive rate versus the true positive rate. As shown in the graph,

[1]Multilabel emotions model modified from github.com/arghyadeep99/Multi-label-Emotion-Classification

the sweep with the highest AUC score had a batch size of 64, dropout rate of 0.3, and a learning rate of 0.00003. We trained the model for 10 epochs and split the data into 80% train, 10% validation, and 10% test.

We ran multiple versions of the model. One version of the model used BERT-base-uncased to encode the stories and then used two linear layers to generate scores which were then passed into argmax to make classifications. A similar version of the model used RoBERTa which we pre-trained on writingPrompts and wikiPlots data. Our full model combines the output of BERT-base-uncased on the text through a linear layer with the word2vec encodings of our comet inferences through a linear layer and our emotions vectors through a linear layer. The concatenation of these outputs is then passed through a linear layer to generate scores for whether the story is coherent or incoherent. Since we did not have COMET inferences for our entire dataset, this version of the model suffered in performance from seeing less data.

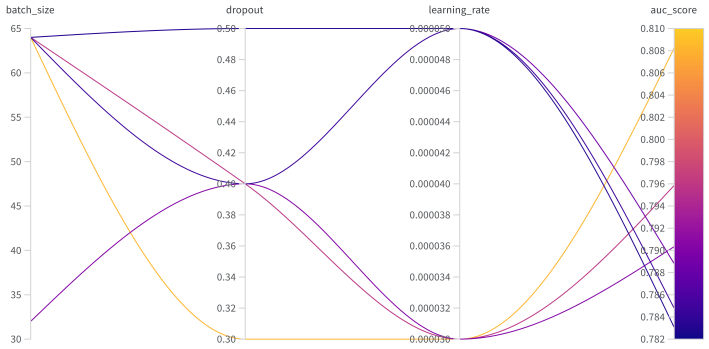

Figure 2: We ran 6 sweeps to tune hyperparameters.

5.4 Results

The multilabel emotion classifier trained on the Commonsense Story dataset achieved a training loss of 0.485, a validation loss of 0.525, and an AUC score of 0.811. On the test data, the AUC score was 0.797.

We also ran a number of experiments as ablation tests to investigate the efficacy of our model. The results are shown below:

BERT version	Data-subset	Accuracy	F1 Score 0/1
BERT-base-uncased	Story Cloze	0.82	0.76/0.82
BERT-base-uncased	ROC/GPT-2	0.96	0.96/0.95
RoBERTa-base-pretrained	Story Cloze	0.61	0.46/0.70
RoBERTa-base-pretrained	ROC/GPT-2	0.83	0.84/0.82
BERT-base-uncased	full data	0.67	0.70/0.61
Full model, 10 epochs	full data	0.63	0.65/0.61
Full model, 10 epochs, tuned layers	full data	0.66	0.67/0.62
Full model, 15 epochs, tuned layers	full data	0.67	0.68/0.63

Table 2: Results of our model's accuracy

As we can see, the model is able to perform slightly better on the kinds of stories that are labeled with the most common label for the dataset that it is trained on: for the ROCStories dataset there were more positive examples even with our data augmentation, and for the fully processed dataset there were more negative examples.

6 Analysis

We hypothesize that the reduction in accuracy from pretraining could possibly be from the length of pretraining (44 hours) or (more likely) from the lower quality of the datasets we used for pretraining (WritingPrompts [5] and WikiPlots[6]). Furthermore we tuned the model's layers after 1 iteration to have fewer entries in the linear layer for the COMET embeddings and emotion vectors since we saw that the BERT encodings seemed to contribute more to the accuracy.

Another factor that may have contributed to our model's lower-than-baseline accuracy may have been the way we decided to process the data generated from COMET. We encoded the words of the COMET outputs with word2vec and computed the centroid of these embeddings which may not have been able to most expressively capture the logical meanings of the inferences (perhaps we could have used a LSTM encoding or used BERT again, especially with attention to the sentences before and after which the COMET relations should have applied to). In addition, using other or more COMET relations could have been the most effective.

7 Conclusion

Comparing the results of our different models and the models used by Mostafazadeh et al. (the highest accuracy achieved being 58.5)%, we noted the success of using BERT-base-uncased with a linear layer for the Story Cloze task. While our data-augmentation strategies were successful in increasing the accuracy of the model, we were unable to outperform BERT by passing additional extracted story features to our model. This is somewhat unsurprising, as empirically, we noted that inferences generated by COMET were often low-quality. In addition, we were hoping that the extracted emotions could model the story-flow (we expect a story with mostly positive emotions to have an ending with positive emotions), having incorrect emotions classifications would disrupt the models ability to make these kinds of judgements. And since our AUC for our emotions model is not approximately 1.00, we would expect some incorrect emotions classifications for our stories, making extracted emotions an unreliable metric for classifying story coherence.

One area for further work could be expanding the domain of the types of stories our model could classify, as our "incoherent" stories all have one logical inconsistency in the ending sentence. Training a BERT-based model on a larger and more diverse dataset of good and bad stories could lead to a more valuable model for evaluating story coherence.

References

[1] Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James 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, 2016.

[2] Siddarth Srinivasan, Richa Arora, and Mark Riedl. A simple and effective approach to the story cloze test. *arXiv preprint arXiv:1803.05547*, 2018.

[3] 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.

[4] Zhongyang Li, Xiao Ding, and Ting Liu. Story ending prediction by transferable BERT. *CoRR*, abs/1905.07504, 2019.

[5] Angela Fan, Mike Lewis, and Yann Dauphin. Hierarchical neural story generation, 2018.

[6] @markriedl. Wikiplots dataset, 2017.