An RNN-based Binary Classifier for the Story Cloze Test

Melissa Roemmele
Institute for Creative Technologies
University of Southern California
roemmele@ict.usc.edu

Sosuke Kobayashi*
Preferred Networks, Inc.
sosk@preferred.jp

Naoya Inoue
Tohoku University
naoya-i@ecei.tohoku.ac.jp

Andrew M. Gordon
Institute for Creative Technologies
University of Southern California
gordon@ict.usc.edu

Abstract

The Story Cloze Test consists of choosing a sentence that best completes a story given two choices. In this paper we present a system that performs this task using a supervised binary classifier on top of a recurrent neural network to predict the probability that a given story ending is correct. The classifier is trained to distinguish correct story endings given in the training data from incorrect ones that we artificially generate. Our experiments evaluate different methods for generating these negative examples, as well as different embedding-based representations of the stories. Our best result obtains 67.2% accuracy on the test set, outperforming the existing top baseline of 58.5%.

1 Introduction

Automatically predicting "what happens next" in a story is an emerging AI task, situated at the point where natural language processing meets commonsense reasoning research. Story understanding began as classic AI planning research (Meehan, 1977, e.g.), and has evolved with the shift to data-driven AI approaches by which large sets of stories can be analyzed from text (Granroth-Wilding and Clark, 2016; Li et al., 2013; McIntyre and Lapata, 2009, e.g.). A barrier to this research has been the lack of standard evaluation schemes for benchmarking progress. The new Story Cloze Test (Mostafazadeh et al., 2016) addresses this need through a binary-choice evaluation format: given the beginning sentences of a story, the task is to choose which of two given sentences best completes the story. The cloze framework also provides training stories (referred to here as the ROC corpus) in the same domain as the evaluation items. Mostafazadeh et al. details the crowdsourced authoring process for this dataset. Ultimately the training data consists of 97,027 five-sentence stories. The separate cloze test has 3742 items (divided equally between validation and test sets) each containing the first four sentences of a story with a correct and incorrect ending to choose from.

In the current paper, we describe a set of approaches for performing the Story Cloze Test. Our best result obtains 67.2% accuracy on the test set, outperforming Mostafazadeh et al.'s best baseline of 58.5%. We first report two additional unsupervised baselines used in other narrative prediction tasks. We then describe our supervised approach, which uses a recurrent neural network (RNN) with a binary classifier to distinguish correct story endings from artificially generated incorrect endings. We compare the performance of this model when alternatively trained on different story encodings and different strategies for generating incorrect endings.

2 Story Representations

We examined two ways of representing stories in our models, both of which encode stories as vectors of real numbers known as embeddings. This was motivated by the top performing baseline in Mostafazadeh et al. which used embeddings to select the candidate story ending with the higher cosine similarity to its context.

Word Embeddings: We first tried encoding stories with word-level embeddings using the word2vec model (Mikolov et al., 2013), which learns to represent words as n-dimensional vector of real values based on neighboring words. We compared two different sets of vectors: 300-

*This research was conducted at his previous affiliation, Tohoku University.

Proceedings of the 2nd Workshop on Linking Models of Lexical, Sentential and Discourse-level Semantics, pages 74–80,
Valencia, Spain, April 3, 2017.

dimension vectors trained on the 100-billion word Google News dataset[1] and 300-dimension vectors that we trained on ROC corpus itself. The latter were trained using the gensim word2vec library[2], with a window size of 10 words and negative sampling of 25 noise words. All other parameters were set to the default values given by the library. By comparing these two sets of embeddings, we intended to determine the extent to which our models can rely only on the limited training data provided for this task. In our supervised experiments we averaged the embeddings of the words in each sentence, resulting in a single vector representation of the entire sentence.

Sentence Embeddings: The second embedding strategy we used was the skip-thought model (Kiros et al., 2015), which produces vectors that encode an entire sentence. Analogous to training word vectors by predicting nearby words, the skip-thought vectors are trained to predict nearby sentences. We evaluated two sets of sentence vectors: 4800-dimension vectors trained on the 11,000 books in the BookCorpus dataset[3], and 2400-dimension vectors we trained ourselves on the ROC corpus[4]. The latter BookCorpus vectors were also used in a baseline that measured vector similarity between the story context and candidate endings in Mostafazadeh et al.

3 Unsupervised Approaches

Mostafazadeh et al. applied several unsupervised baselines to the Story Cloze Test. We evaluated two additional approaches due to their success on other narrative prediction tasks.

Average Maximum Similarity (AveMax): The AveMax model is a slight variation on Mostafazadeh et al.'s averaged word2vec baseline. It is currently implemented to predict story continuations from user input in the recently developed DINE application[5]. Instead of selecting the embedded candidate ending most similar to the context, this method iterates through each word in the ending, finds the word in the context with most similar embedding, and then takes the mean of these maximum similarity embeddings. We evaluated this method using both the word embeddings from the Google News dataset and the ROC corpus.

Pointwise Mututal Information (PMI): The PMI model was used successfully on the Choice of Plausible Alternatives task (COPA) (Roemmele et al., 2011; Gordon et al., 2011; Gordon et al., 2012; Luo et al., 2016) which similarly to the Story Cloze Test uses a binary-choice format to elicit inferences about a segment of narrative text. This model relies on lexical co-occurrence counts (of raw words rather than embeddings) to compute a 'causality score' about how likely one sentence is to follow another in a story. We applied the same approach to the Story Cloze Test to select the final sentence with the higher causality score of the two candidates. We evaluated word counts from two different sources: a corpus of one million stories extracted from personal weblogs (as was used in Gordon et al.) and the ROC corpus.

4 Supervised Approaches

Given the moderate size of the ROC corpus at almost 100,000 stories, and that the Story Cloze Test can be viewed as a classification task choosing from two possible outputs, we investigated a supervised approach. Unlike the training data for traditional classification models, the ROC corpus does not involve a set of discrete categories by which stories are labeled. Moreover, while the Story Cloze Test provides a correct and incorrect outcome to choose from, the training data only contains the correct ending for a given story. So our strategy was to create a new training set with binary labels of 1 for correct endings (positive examples) and 0 for incorrect endings (negative examples). Each story in the corpus was considered a positive example. Given a positive example, we generated a negative example by replacing its final sentence with an incorrect ending. As described below, we generated more than one negative ending per story, so that each positive example had multiple negative counterparts. Our methods for generating negative examples are described in the next section. Our approach was to train a binary classifier to distinguish between these positive and negative examples.

The binary classifier is integrated with an RNN. RNNs have been used successfully for other narrative modeling tasks (Iyyer et al., 2016; Pichotta and Mooney, 2016). Our model takes the context sentences and ending for a particular story as in-

[1] https://code.google.com/archive/p/word2vec/

[2] https://radimrehurek.com/gensim/models/word2vec.html

[3] https://github.com/ryankiros/skip-thoughts

[4] We used the same code and default parameters available at the above GitHub page.

[5] http://dine.ict.usc.edu

Context	Correct	Type	Incorrect
Hal was walking his dog one morning. A cat ran across their path. Hal's dog strained so hard, the leash broke! He chased the cat for several minutes.	Finally Hal lured him back to his side.	**Rand**	Tom was kicked out of the game.
		Back	A cat ran across their path.
		Near	His dog had to wear a leg cast for weeks.
		Near	His dog is too fast and runs off.
		Near	Rod realized he should have asked before petting the dog.
		LM	When she woke up, she realized he had no dog noises.
		LM	When he got to the front, he saw a dog, squirrel, and dog.
		LM	When he got to the front office, he found a cat in the ditch.
John woke up sick today. He washed his face in the bathroom. John went into the kitchen to make some soup. He put a bowl of soup into the microwave.	John dropped the soup when he grabbed it from the microwave.	**Rand**	She waited for months for her hair to grow back out.
		Back	He put a bowl of soup into the microwave.
		Near	Dan returned to the couch and watched a movie with his snack.
		Near	The doctor gave him medicine to get better.
		Near	Finally, he ate it.
		LM	He brushed his teeth and ate it for a while, he was sad.
		LM	He put the bowl in his microwave, and went to the kitchen.
		LM	He brushed her teeth, but the candles didn't feel so he didn't have any.

Table 1: Examples of generated negative endings

put and then returns the probability of that ending being correct, using the ending labels as feedback during training. Specifically, we combine the sentence representations of the context and final sentences into one sequence and feed each sentence as a timestep into a single 1000-node GRU (Cho et al., 2014) hidden layer. The values of the final hidden state are given to a top feed-forward layer composed of one node with sigmoid activation. A binary cross-entropy objective function is applied to train the network to maximize the probability of positive examples being correct. All experiments used RMSprop (Hinton et al., 2012) with a batch size of 100 to optimize the model over 10 training epochs. After training, given a cloze test item, the model predicted a probability score for each candidate ending, and the ending with the higher score was selected as the response for that item.

5 Incorrect Ending Generation

We examined four different ways to generate the incorrect endings for the classifier. Table 1 shows examples of each.

Random (Rand): First, we simply replaced each story's ending with a randomly selected ending from a different story in the training set. In most cases this ending will not be semantically related to the story context, so this approach would be expected to predict endings based strictly on semantic overlap with the context.

Backward (Back): The Random approach generates negative examples in which the semantics of the context and ending are most often far apart. However, these examples may not represent the items in the Story Cloze Test, where the endings generally both have some degree of semantic coherence with the context sentences. To generate negative examples in the same semantic space as the correct ending, we replaced the fifth sentence of a given story with one of its four context sentences (i.e. a backward sentence). This results in an ending that is semantically related to the story, but is typically incoherent given its repetition in the story.

Nearest-Ending (Near): The Nearest-Ending approach aims to find endings that are very close to the correct ending by using an ending for a similar story in the corpus. Swanson and Gordon (2012) presented this model in their interactive storytelling system. Given a story context, we

retrieved the most similar story in the corpus (in terms of cosine similarity), and then projected the final sentence of the similar story as the ending of the given story. Multiple endings were produced by finding the N most similar stories. The negative examples generated by this scheme can be seen as 'almost' positive examples with likely coherence errors, given the sparsity of the corpus. This is in line with the cloze task where both endings are plausible, but the correct answer is more likely than the other.

Language Model (LM): Separate from the binary classifier, we trained an RNN-based language model (Mikolov et al., 2010) on the ROC corpus. The LM learns a conditional probability distribution indicating the chance of each possible word appearing in a sequence given the words that precede it. During training, the LM iterated through a story word by word, each time updating its predicted probability of the next observed word. During generation, we gave the LM the context of each training story and had it produce a final sentence by sampling words one by one according to the predicted distribution, as described in Sutskever et al. (2011). Multiple sentences were generated for the same story by sampling the N most probable words at each timestep. The LM had a 200-node embedding layer and two 500-node GRU layers, and was trained using the Adam optimizer (Kingma and Ba, 2015) with a batch size of 50. This approach has an advantage over the Nearest-Ending method in that it leverages all the stories in the training data for generation, rather than predicting an ending based on a single story. Thus, it can generate endings that are not directly observed in the training corpus. Like the nearest-ending approach, an ideal LM would be expected to generate positive examples similar to the original stories it is trained on. However, we found that the LM-generated endings were relevant to the story context but had less of a commonsense interpretation than the provided endings, again likely due to training data sparsity.

6 Experiments

We trained a classifier for each type of negative ending and additionally for each type of embedding, shown in Table 2. For each correct example, we generated multiple incorrect examples. We found that setting the number of negative samples per positive example near 6 produced the best results on the validation set for all configurations, so we kept this number consistent across experiments. The exception is the Backward method, which can only generate one of the first four sentences in each story. For each generation method, the negative samples were kept the same across runs of the model with different embeddings, rather than re-sampling for each run. After discovering that our best validation results came from the random endings, we also evaluated combinations of these endings with the other types to see if they could further boost the model's performance. The samples used by these combined-method experiments were a subset of the negative samples generated for the single-method results.

Table 2 shows the accuracy of all unsupervised and supervised models on both the validation and test sets, with the best test result within each group in bold. Among the unsupervised models, the AveMax model with the GoogleNews embeddings (55.2% test accuracy) performs comparably to Mostafazadeh et al.'s word2vec similarity model (53.9%). The PMI approach performs at the same level as the current best baseline of 58.5%, and the counts from the ROC stories are just as effective (59.9%) as those from the much larger blog corpus (59.1%).

The best test result using the GoogleNews word embeddings (61.5%) was slightly better than that of the ROC word embeddings (58.8%). Among the single-method results, the word embeddings were outperformed by the best result of the skip-thought embeddings (63.2%), suggesting that the skip-thought model may capture more information about a sentence than simply averaging its word embeddings. For this reason we skipped evaluating the word embeddings for the combined-ending experiments. One caveat to this is the smaller size of the word embeddings relative to the skip-thought vectors. While it is unusual for word2vec embeddings to have more than a thousand dimensions, to be certain that the difference in performance was not due to the difference in dimensionality, we performed an ad-hoc evaluation of word embeddings that were the same size as the ROC sentence vectors (2400 nodes). We computed these vectors from the ROC corpus in the same way described in Section 2, and applied them to our best-performing data configuration (Rand-3 + Back-1 + Near-1 + LM-1). The result (57.9%) was still lower than that produced by the cor-

	Val	Test
Unsupervised		
AveMax		
GoogleNews WordEmb	0.553	**0.552**
ROC WordEmb	0.548	0.547
PMI		
Blog Corpus	0.585	0.591
ROC Corpus	0.581	**0.599**
Supervised		
Rand-6		
GoogleNews WordEmb	0.625	0.585
ROC WordEmb	0.605	0.584
BookCorpus SentEmb	0.645	**0.632**
ROC SentEmb	0.639	0.631
Back-4		
GoogleNews WordEmb	0.529	0.540
ROC WordEmb	0.528	0.553
BookCorpus SentEmb	0.545	0.539
ROC SentEmb	0.548	**0.560**
Near-6		
GoogleNews WordEmb	0.641	0.615
ROC WordEmb	0.585	0.588
BookCorpus SentEmb	0.649	**0.621**
ROC SentEmb	0.632	0.615
LM-6		
GoogleNews WordEmb	0.524	0.534
ROC WordEmb	0.523	**0.544**
BookCorpus SentEmb	0.520	0.507
ROC SentEmb	0.514	0.512
Rand-4 + Back-2		
BookCorpus SentEmb	0.662	**0.669**
ROC SentEmb	0.664	0.664
Rand-4 + Near-2		
BookCorpus SentEmb	0.636	**0.641**
ROC SentEmb	0.650	0.609
Rand-4 + LM-2		
BookCorpus SentEmb	0.624	0.607
ROC SentEmb	0.640	**0.653**
Rand-3 + Back-1 + Near-1 + LM-1		
ROC WordEmb (2400)	0.599	0.579
BookCorpus SentEmb	0.656	**0.672**
ROC SentEmb	0.680	0.661

Table 2: Accuracy on the Story Cloze Test

responding ROC sentence embeddings (66.1%), supporting our idea that the skip-thought embeddings are a better sentence representation. Interestingly, though the BookCorpus sentence vectors obtained the best result overall (67.2%), they performed on average the same as the ROC ones (mean accuracy of 61.1% versus 61.3%, respectively), despite that the former have more dimensions (4800) and were trained on several more stories. This might suggest it helps to model the unique genre of stories contained in the ROC corpus for this task.

The best results in terms of data generation incorporate the Random endings, suggesting that for many of the items in the Story Cloze Test, the correct ending is the one that is more semantically similar to the context. Not surprisingly, the Backward endings have limited effect on their own (best result 56%), but they boost the performance of the Random endings when combined (best result 66.9%). We expected that the Nearest-Ending and LM endings would have an advantage over the Random endings, but our results didn't show this. The best result for the Nearest-Ending method was 62.1% compared to 63.2% produced by the Random endings. The LM endings fared particularly badly on their own (best result 54.4%). We noticed the LM seemed to produce very similar endings across different stories, which possibly influenced this result. The best result overall (67.2%) was produced by the model that sampled from all four types of endings, though it was only trivially higher than the best result for the combined Random and Backward endings (66.9%). Still, we see opportunity in the technique of using generative methods to expand the training set. We only generated incorrect endings in this work, but ideally this approach could generate correct endings as well, given that a story has multiple possible correct endings. It is possible that the small size of the ROC corpus limited our current success with this idea, so in the future we plan to pursue this using a much larger story dataset.

7 Acknowledgments

The projects or efforts depicted were or are sponsored by the U. S. Army. The content or information presented does not necessarily reflect the position or the policy of the Government, and no official endorsement should be inferred. Additionally, this work was partially supported by JSPS KAKENHI Grant Numbers 15H01702, 16H06614, and CREST, JST.

References

Kyunghyun Cho, Bart van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder-decoder approaches. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 103–111, Doha, Qatar, October. Association for Computational Linguistics.

Andrew S. Gordon, Cosmin Adrian Bejan, and Kenji Sagae. 2011. Commonsense causal reasoning using millions of personal stories. In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence*, AAAI'11, pages 1180–1185. AAAI Press.

Andrew S. Gordon, Zornitsa Kozareva, and Melissa Roemmele. 2012. Semeval-2012 task 7: Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics - Volume 1: Proceedings of the Main Conference and the Shared Task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*, SemEval '12, pages 394–398, Stroudsburg, PA, USA. Association for Computational Linguistics.

Mark Granroth-Wilding and Stephen Clark. 2016. What Happens Next? Event Prediction Using a Compositional Neural Network Model. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, pages 2727–2733. AAAI Press.

Geoffrey Hinton, Nitish Srivastava, and Kevin Swersky. 2012. Overview of mini-batch gradient descent. `http://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf`.

Mohit Iyyer, Anupam Guha, Snigdha Chaturvedi, Jordan Boyd-Graber, and Hal Daumé III. 2016. Feuding families and former friends: Unsupervised learning for dynamic fictional relationships. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1534–1544, San Diego, California, June. Association for Computational Linguistics.

Diederik Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *The International Conference on Learning Representations (ICLR)*, San Diego, May.

Ryan Kiros, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 3294–3302. Curran Associates, Inc.

Boyang Li, Stephen Lee-Urban, George Johnston, and Mark O. Riedl. 2013. Story generation with crowdsourced plot graphs. In *Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence*, AAAI'13, pages 598–604. AAAI Press.

Zhiyi Luo, Yuchen Sha, Kenny Q. Zhu, Seung-won Hwang, and Zhongyuan Wang. 2016. Commonsense causal reasoning between short texts. In *Proceedings of the Fifteenth International Conference on Principles of Knowledge Representation and Reasoning*, KR'16, pages 421–430. AAAI Press.

Neil McIntyre and Mirella Lapata. 2009. Learning to tell tales: A data-driven approach to story generation. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 217–225, Suntec, Singapore, August. Association for Computational Linguistics.

James R. Meehan. 1977. Tale-spin, an interactive program that writes stories. In *Proceedings of the 5th International Joint Conference on Artificial Intelligence - Volume 1*, IJCAI'77, pages 91–98, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Tomas Mikolov, Martin Karafiat, Lukas Burget, Jan Cernocky, and Sanjeev Khudanpur. 2010. Recurrent Neural Network based language model. In *Proceedings of the 11th Annual Conference of the International Speech Communication Association*, INTERSPEECH 2010, pages 1045–1048.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems*, NIPS'13, pages 3111–3119, USA. Curran Associates Inc.

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. 2016. 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. Association for Computational Linguistics.

Karl Pichotta and Raymond J. Mooney. 2016. Learning statistical scripts with lstm recurrent neural networks. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, pages 2800–2806. AAAI Press.

Melissa Roemmele, Cosmin Adrian Bejan, and Andrew S. Gordon. 2011. Choice of Plausible Alternatives : An Evaluation of Commonsense Causal Reasoning. *AAAI Spring Symposium: Logical Formalizations of Commonsense Reasoning*, pages 90–95, March.

Ilya Sutskever, James Martens, and Geoffrey Hinton. 2011. Generating text with recurrent neural networks. In Lise Getoor and Tobias Scheffer, editors, *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, ICML '11, pages 1017–1024, New York, NY, USA, June. ACM.

Reid Swanson and Andrew S. Gordon. 2012. Say anything: Using textual case-based reasoning to enable open-domain interactive storytelling. *ACM Trans. Interact. Intell. Syst.*, 2(3):16:1–16:35, September.