Story Cloze task – XXX System

Anonymous EACL submission

Abstract

This paper describes our system for the Linking Models of Lexical, Sentential and Discourse-level Semantics (LSDSem 2017) shared task – the *Story Cloze Task*. Our system feeds a linear classifier with a variety of features, including both the output of a neural language model and style features. We report 75.2% accuracy on the task.

1 Introduction

As an effort to advance commonsense understanding, Mostafazadeh et al. (2016) developed the *story cloze task*, which is the focus of the LSDSem 2017 shared task. In this task, systems are given two short, self-contained stories, which differ only in their last sentence: one has a *right* (coherent) ending, and the other has a *wrong* (incoherent) ending. The task is to tell which is the *right* story. In addition to the task, the authors also introduced the *ROC story corpus* – a training corpus of five-sentence (coherent) stories. Table 1 shows an example of a *coherent* story and an *incoherent* story from the story cloze task.

In this paper, we describe the system we submitted for the shared task. Our system explores several types of features for the task. First, we train a neural language model (Mikolov et al., 2010) on the ROC story corpus. We use the probabilities assigned by the model to each of the endings (*right* and *wrong*) as classification features.

Second, we attempt to distinguish between *right* and *wrong* endings using style features, such as sentence length, character n-grams and word n-grams. Our intuition is that the *right* endings use a different style compared to the *wrong* endings. The features we use are motivated by previous works on tasks such as age (Schler et al., 2006), gender (Argamon et al., 2003), and authorship profiling (Stamatatos, 2009), for which similar style features are known to be very effective.

Story Prefix	Ending
Kathy went shopping. She found a pair of great shoes. The shoes were $300. She bought the shoes.	She felt buyer's remorse after the purchase.
	Kathy hated buying shoes.

Table 1: Examples of stories from the story cloze task (Mostafazadeh et al., 2016). The left column shows that first four sentences of a story. The right column shows two contrastive endings for the story: a coherent ending and a incoherent one.

We feed our features to a logistic regression classifier, and evaluate our system on the shared task. Our system obtains 75.2% accuracy on the test set.

2 System Description

We design a system that predicts, given a pair of story endings, which is the *right* one and which is the *wrong* one. Our system applies a linear classifier guided by several types of features to solve the task. We describe the system in detail below.

2.1 Model

We train a binary logistic regression classifier to distinguish between *right* and *wrong* stories. We use the set of *right* stories as positive samples and the set of *wrong* stories as negative samples. At test time, for a given pair, we consider the classification results of both candidates. If our classifier assigns different labels to each candidate, we keep them. If not, the label whose posterior probability is lower is reversed. We describe the classification

features below.

2.2 Features

We use two types of features, designed to capture different aspects of the problem. We use *neural language model* features to leverage corpus level word distributions, specifically longer term sequence probabilities. We use *stylistic* features to capture differences in writing between *coherent* story endings and *incoherent* ones.

Language Model Features. We experiment with state-of-the-art text comprehension models, specifically an LSTM (Hochreiter and Schmidhuber, 1997) recurrent neural network language model (RNNLM; Mikolov et al. (2010)). Our RNNLM is used to generate two different probabilities: $p_\theta(\text{ending})$, which is the language model probability of the fifth sentence alone and $p_\theta(\text{ending} \mid \text{story})$, which is the RNNLM probability of the fifth sentence given the first four sentences. We use both of these probabilities as classification features.

In addition, we also apply a third feature:

$$\frac{p_\theta(\text{ending} \mid \text{story})}{p_\theta(\text{ending})} \quad (1)$$

The intuition is that a *correct* ending should be unsurprising (to the model) given the four preceding sentences of the story (the numerator), controlling for the inherent surprise of the words in that ending (the denominator).

Stylistic Features. We hypothesize that *right* and *wrong* endings might be distinguishable using style features. We adopt features that have been shown useful in the past in tasks such as detection of age (Schler et al., 2006; Rosenthal and McKeown, 2011; Nguyen et al., 2011), gender (Argamon et al., 2003; Schler et al., 2006; Bamman et al., 2014), and native language (Koppel et al., 2005; Tsur and Rappoport, 2007; Bergsma et al., 2012).

We add the following features to capture style differences between the two endings. These features are computed on the story endings alone, and do not consider the first four (shared) sentences of each story.

- ***Length***. The number of words in the sentence.
- ***Word n-grams.*** We use sequences of 1-5 words. Following (Tsur et al., 2010; Schwartz et al., 2013), we distinguish between high frequency and low frequency words. Specifically, we replace content words, which are often low frequency, with their part-of-speech tags (Nouns, Verbs, Adjectives and Adverbs).
- ***Character n-grams.*** Character n-grams are useful features in the detection of author style (Stamatatos, 2009) or language identification (Lui and Baldwin, 2011). We use character 4-grams.

2.3 Experimental Setup

We use Python's sklearn logistic regression implementation with L_2 regularization, performing grid search on the development set to tune a single hyperparameter – the regularization parameter.

We train the RNNLM using a single-layer LSTM of hidden dimension 512. We use the ROC Stories for training, setting aside 10% for validation of the language model.[1] We replace all words occurring less than 3 times by a special out-of-vocabulary character, yielding a vocabulary size of 21,582. Only during training, we apply a dropout rate of 60% while running the LSTM over all 5 sentences of the stories. Using AdamOptimizer (Kingma and Ba, 2014) and a learning rate of $\eta = .001$, we train to minimize cross-entropy.

For computing the style features, we keep n-gram (character or word) features that occur at least five times in the training set. All stylistic feature values are normalized to [0-1]. For the POS features, we tag all endings with the Spacy POS tagger.[2] The total number of features used by our system is 7,651.

3 Results

The performance of our system is described in Table 2. With 75.2% accuracy, our system achieves 15.3% better than the published state of the art (Salle et al., 2016). The table also shows an analysis of the different features types used by our system. While our RNNLM features alone reach 67.7%, the style features perform better – 72.4%. This suggests that while this task is about story understanding, there is some information contained in stylistic features, which are slightly less sensitive to content. As expected, the RNNLM fea-

[1] We train on both the Spring 2016 and the Winter 2017 datasets, a total of roughly 100K stories

[2] `spacy.io/`

Model	Accuracy
DSSM (Mostafazadeh et al., 2016)	0.585
LexVec (Salle et al., 2016)	0.599
RNNLM Features	0.677
Stylistic features	0.724
Combined (Style + RNNLM)	**0.752**
Human judgment	1.000

Table 2: Results on the test set of the story cloze task. The first block are published results, the second block are our results. LexVec results are taken from (Speer et al., 2016). Human judgement scores are taken from (Mostafazadeh et al., 2016).

tures complement the stylistic ones, boosting performance by 7.5% (over the RNNLM) and 2.8% (over the style features).

4 Conclusion

This paper described our submission to the LSDSem 2017 Shared Task. Our system leveraged both neural language model features and stylistic features, achieving 75.2% accuracy on the classification task.

References

Shlomo Argamon, Moshe Koppel, Jonathan Fine, and Anat Rachel Shimoni. 2003. Gender, genre, and writing style in formal written texts. *TEXT-THE HAGUE THEN AMSTERDAM THEN BERLIN-*, 23(3):321–346.

David Bamman, Jacob Eisenstein, and Tyler Schnoebelen. 2014. Gender identity and lexical variation in social media. *Journal of Sociolinguistics*, 18(2):135–160.

Shane Bergsma, Matt Post, and David Yarowsky. 2012. Stylometric analysis of scientific articles. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 327–337, Montréal, Canada, June. Association for Computational Linguistics.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Moshe Koppel, Jonathan Schler, and Kfir Zigdon. 2005. Determining an author's native language by mining a text for errors. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 624–628. ACM.

Marco Lui and Timothy Baldwin. 2011. Cross-domain feature selection for language identification. In *Proceedings of 5th International Joint Conference on Natural Language Processing*. Citeseer.

Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *Interspeech*, volume 2, page 3.

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.

Dong Nguyen, Noah A. Smith, and Carolyn P. Rosé. 2011. Author age prediction from text using linear regression. In *Proceedings of the ACL Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities*.

Sara Rosenthal and Kathleen McKeown. 2011. Age prediction in blogs: A study of style, content, and online behavior in pre- and post-social media generations. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 763–772, Portland, Oregon, USA, June. Association for Computational Linguistics.

Alexandre Salle, Marco Idiart, and Aline Villavicencio. 2016. Enhancing the lexvec distributed word representation model using positional contexts and external memory. *arXiv preprint arXiv:1606.01283*.

Jonathan Schler, Moshe Koppel, Shlomo Argamon, and James W Pennebaker. 2006. Effects of age and gender on blogging. In *AAAI spring symposium: Computational approaches to analyzing weblogs*, volume 6, pages 199–205.

Roy Schwartz, Oren Tsur, Ari Rappoport, and Moshe Koppel. 2013. Authorship attribution of micromessages. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1880–1891, Seattle, Washington, USA, October. Association for Computational Linguistics.

Robert Speer, Joshua Chin, and Catherine Havasi. 2016. Conceptnet 5.5: An open multilingual graph of general knowledge. *arXiv preprint arXiv:1612.03975*.

Efstathios Stamatatos. 2009. A survey of modern authorship attribution methods. *Journal of the American Society for information Science and Technology*, 60(3):538–556.

Oren Tsur and Ari Rappoport. 2007. Using classifier features for studying the effect of native language on the choice of written second language words. In *Proceedings of the Workshop on Cognitive Aspects of Computational Language Acquisition*, pages 9–16, Prague, Czech Republic, June. Association for Computational Linguistics.

Oren Tsur, Dmitry Davidov, and Ari Rappoport. 2010. Icwsm-a great catchy name: Semi-supervised recognition of sarcastic sentences in online product reviews. In *ICWSM*, pages 162–169.