Who's to say what's funny? A computer using Language Models and Deep Learning, that's who!

Anonymous ACL submission

Abstract

Humor is a defining characteristic of human beings. Our goal is to develop methods that automatically detect humorous statements and rank them on a continuous scale. In this paper we report on results using a Language Model approach, and outline our plans for using methods from Deep Learning.

1 Introduction

Computational humor is an emerging area of research that ties together ideas from psychology, linguistics, and cognitive science. *Humor generation* is the problem of automatically creating humorous statements (e.g., (Stock and Strapparava, 2003), (Özbal and Strapparava, 2012)) while *humor detection* is concerned with identifying humor in text, and is sometimes treated as a binary classification problem where the goal is to decide if some input is humorous or not (e.g., (Mihalcea and Strapparava, 2006), (Zhang and Liu, 2014), (Shahaf et al., 2015), (Miller and Gurevych, 2015)). However, our interest is more in the continuous and subjective nature of humor.

In order to account for the subjective nature of humor, we learn a sense of humor from a dataset of tweets geared towards a particular style of humor (Potash et al., 2016). This data consists of humorous tweets which have been submitted in response to hashtags given during the Comedy Central TV show *@midnight with Chris Hardwick*. To represent the continuous aspect of humor (where not all funny jokes are equally amusing), we focus on using Language Models and methods from Deep Learning that allow for the ranking of potentially humorous statements relative to each other.

2 Language Models

We start with traditional Ngram language models for two reasons:

1. Ngram language models can be customized to a particular kind of humor by using examples of that as the training data for the model, and

2. Ngram language models provide a probability value for each input they are given, thus making it possible to rank statements relative to each other.

We began this research by participating in SemEval-2017 Task 6 #HashtagWars: Learning a Sense of Humor (Potash et al., 2017). This included two subtasks : Pairwise Comparison (Subtask A) and Semi-ranking (Subtask B). Pairwise comparison asks a system to choose the funnier of two tweets, while semi-ranking requires that 10 tweets be placed in three categories (most funny, funny, less funny).

Our system estimated tweet probabilities using Ngram language models. We created models from two different data sets - a collection of funny tweets from the @midnight program, and a corpus of news data that is freely available for research[1]. We scored tweets using each model. Tweets that have a higher probability when using the funny tweet model were considered funnier. However, tweets that have a lower probability according to the news language model are viewed as being funnier since they are least like the (unfunny) news corpus. We took a fairly standard approach to language modeling and used bigrams and trigrams as features in the model. We

[1] http://www.statmt.org/wmt11/featured-translation-task.html

used KenLM (Heafield et al., 2013) as our language modeling too, and used modified Kneser-Ney smoothing and back-off techniques. More details about our system refers to

Table 1 shows our results for both data sets when trained on bigrams and trigrams. The accuracy and the distance measures are defined by the task organizers (Potash et al., 2017). We seek high accuracy in picking the funnier tweet (subtask A) and low distance (from the gold standard) in organizing the tweets into categories (subtask B).

DataSet	N-gram	Subtask A Accuracy	Subtask B Distance
tweets	trigram	0.397	0.967
tweets	bigram	0.406	0.944
news	trigram	0.627	0.872
news	bigram	0.624	0.853

Table 1: Experimental results

These results show that the trigram model trained on the news data has a significant advantage over the tweets data. Bigram language models performed slightly better than trigram models on both data sets. In the official evaluation of SemEval-2017 Task 6 our most effective method was trigram language models trained on the news data - this placed fourth in Subtask A and first in Subtask B.

We believe that the significant advantage of the news data over the tweet data is caused by the difference in quantity between corpora. The tweet data only consisted of approximately 21,000 tweets, whereas the news data totals approximately 6.2 GB of text. In the future we intend to collect more tweet data, especially those participating in the ongoing #HashtagWars staged nightly by @midnight. We also plan to experiment with equal amounts of tweet data and news data, to see if one has an inherent advantage over the other.

Finally, our language models performed better on subtask A, the pairwise comparison, suggesting that the ranking problem is more demanding and will require further development. To that end we are considering the use of neural network and Deep Learning approaches, which we will discuss in the following section.

3 Deep Learning

Our system performed well in the SemEval task, confirming the ability of language models to detect humor with focusing on its continuous and subjective nature. Going forward, language models certainly can characterize humor in a more comprehensive way with more complex implementations.

Recently, studies have shown that Recurrent Neural Networks (RNN) such as LSTM neural networks are exceptionally powerful in language modeling based on its ability to take into account of all preceding words over a word sequence (Sundermeyer et al., 2012) (Sundermeyer et al., 2015). Also, in their work of learning a sense of humor, (Potash et al., 2016) points out that due to the high portion of out of vocabulary (OOV) words generated by puns in the humorous tweet data, character-level Constitutional Neural Networks (CNN) model would be more suitable for capturing the single-token puns comparing to token level models. In addition, (Kim et al., 2015)'s study proves that character-level neural language models outperform state-of-art word-level neural language models. Moving forward, considering the promising future of neural networks, we would like to build a character-level neural language model (NLM) which relies on both LSTM RNN and character-level CNN to improve our current system performance.

Last but not the least, (Potash et al., 2016) states that external knowledge is crucial in order to detecting humor from the tweet dataset based on its nature. In the future we intend to make use of external knowledge in the NLM in possibly two ways: With a deeper understanding of what kinds of external knowledge are the most useful such as movie and book titles, celebrity and song names, one is to incorporate them as features in the NLM; The other is to combine the NLM with a Neural Turing Machine (NTM) (Graves et al., 2014), which is particularly designed for interacting with external interfaces, with a profound and solid study of NTM.

4 Conclusion

Humor has not been addressed broadly in current computational research area. Our research focuses on humor detection, developing systems that can capture its continuous and subjective nature.

References

Alex Graves, Greg Wayne, and Ivo Danihelka. 2014. Neural turing machines. *arXiv preprint arXiv:1410.5401* .

Kenneth Heafield, Ivan Pouzyrevsky, Jonathan H. Clark, and Philipp Koehn. 2013. Scalable modified Kneser-Ney language model estimation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*. Sofia, Bulgaria, pages 690–696.

Yoon Kim, Yacine Jernite, David Sontag, and Alexander M Rush. 2015. Character-aware neural language models. *arXiv preprint arXiv:1508.06615* .

Rada Mihalcea and Carlo Strapparava. 2006. Learning to laugh (automatically): Computational models for humor recognition. *Computational Intelligence* 22(2):126–142.

Tristan Miller and Iryna Gurevych. 2015. Automatic disambiguation of english puns. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Association for Computational Linguistics, Beijing, China, pages 719–729.

Gözde Özbal and Carlo Strapparava. 2012. A computational approach to the automation of creative naming. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*. Association for Computational Linguistics, pages 703–711.

Peter Potash, Alexey Romanov, and Anna Rumshisky. 2016. # hashtagwars: Learning a sense of humor. *arXiv preprint arXiv:1612.03216* .

Peter Potash, Alexey Romanov, and Anna Rumshisky. 2017. SemEval-2017 Task 6: #HashtagWars: learning a sense of humor. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*. Vancouver, BC.

Dafna Shahaf, Eric Horvitz, and Robert Mankoff. 2015. Inside jokes: Identifying humorous cartoon captions. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, New York, NY, USA, KDD '15, pages 1065–1074.

Oliviero Stock and Carlo Strapparava. 2003. Getting serious about the development of computational humor. In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence*. Acapulco, pages 59–64.

Martin Sundermeyer, Hermann Ney, and Ralf Schlüter. 2015. From feedforward to recurrent lstm neural networks for language modeling. *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)* 23(3):517–529.

Martin Sundermeyer, Ralf Schlüter, and Hermann Ney. 2012. Lstm neural networks for language modeling. In *Interspeech*. pages 194–197.

Xinru Yan and Ted Pedersen. 2017. Duluth at semeval-2017 task 6: Language models in humor detection. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*. Vancouver, BC.

Renxian Zhang and Naishi Liu. 2014. Recognizing humor on twitter. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*. ACM, New York, NY, USA, CIKM '14, pages 889–898.