Extended Abstract for WiNLP

Anonymous ACL submission

Abstract

Humor represents one of the most unique and intelligent activities that define humans. Our research focuses on humor detection by using a variety of methods in order to learn a sense of humor. So far we have developed system Duluth using N-gram language models to recognize humorous tweets, which participated in SemEval-2017 Task 6 and ranked highly in the task evaluation. This paper presents the current work of our research along with promising results, as well as possible future work.

1 Introduction

Humor is considered to be a human-only trait and one of the most amusing and mystifying human activities. It enters the domain of philosophy, sociology, psychology, linguistics and computer science. With the increasing development of Artificial Intelligence(AI), Machine Learning(ML) and computational linguistics, *Computational humor* has found its way to numerous studies. Humor generation has been a prevailing focus of computational humor (e.g., (Stock and Strapparava, 2003), (Özbal and Strapparava, 2012)). However, *humor detection* remains a less explored and challenging problem (e.g., (Mihalcea and Strapparava, 2006), (Zhang and Liu, 2014), (Shahaf et al., 2015), (Miller and Gurevych, 2015)). In our research, we implement systems that try to utilize and combine diverse methods to recognize humor better.

To get started and build a solid foundation of our work, we choose to use N-gram Language Models (LMs) first to tackle the problem. The idea of using language model is to learn a sense of humor by gaining useful information from a word and its neighbors (Jurafsky and Martin, 2009). Our research is also associated with the SemEval-2017 Task 6 #HashtagWars: Learning a Sense of Humor (Potash et al., 2017). The task aims to characterize the sense of humor of a particular source consisting of humorous tweets submitted to a comedy show *@midnight*. There are two subtasks involved: Pairwise Comparison (Subtask A) and Semi-ranking (Subtask B). Our system ranks tweets based on how funny they are by training N-gram LMs on two different corpora, the funny tweets corpus which is provided by the task and the news corpus which is freely available for research.

In order to evaluate how funny a tweet is, we train language models on the tweet data and the news data respectively. Tweets that have a higher probability according to the tweet data language model are ranked as being funnier. However, tweets that are less probable according to the news language model are considered the funnier since they are the least like the (unfunny) news corpus. We rely on both bigrams and trigrams when training our models. We use KenLM (Heafield et al., 2013) as our language modeling tool with modified Kneser-Ney smoothing and back-off technique.

2 Method

Our system Duluth [1] estimated tweet probability using N-gram LMs. First, our system combined all training data into one single file with data preprocessing steps including filtering and tokenization. Second, the system built N-gram language model using KenLM. Then the system computed log probability for each tweet based on the trained N-gram language model. Last but the least is

[1] https://xinru1414.github.io/HumorDetection-SemEval2017-Task6/

the tweet prediction: for Subtask A, given two tweets, the system predicted which one is funnier according to their probability scores; for Subtask B, given a set of tweets associated with one hashtag, the system ranked tweets from the funniest to the least funny according to their probability scores. Note that the system went through these steps on both training datasets respectively.

3 Results

In this section we present the results by using N-gram language model approach. We include results based on language models trained on both datasets using bigram and trigram. Note that the accuracy and the distance measurements listed are defined by the task organizers (Potash et al., 2017). For Subtask A, higher accuracy score represents better system performance. However for Subtask B, lower distance score means the system works more effective.

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: Results based on bigram and trigram LMs on both datasets. The trigram LM trained on the news data ranked 4th place on Subtask A and 1st place on Subtask B during the SemEval task evaluation.

Results show that when comparing datasets, models trained on the news data had a significant advantage over models trained on the tweets data. Moreover, generally speaking bigram models performed slightly better than trigram models on both datasets. The results also show that our system had a better performance on Subtask A over Subtask B.

4 Discussion and Future Works

Our system performed well in SemEval-2017 Task 6, which indicates that simple and straightforward methods such as language model approach could be suitable for hard tasks like humor detection. Going forward, language models certainly could be capable of capturing and representing humor in a more comprehensive manor with more complex implementations. It represents a promising start of our research.

We relied on bigram and trigram language models considering the nature of tweets. They are often short and concise, carrying information within few words. The subtle advantage of bigram language models over trigram language models suggests that moving forward we should consider to use both unigram and character–level language models.

Moreover, 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 was significantly smaller than the news data. In the future we intend to collect more tweet data, especially the ones participating the #HashtagWars. We also plan to experiment on cutting the amount of news data and then build the models to see how the results compare.

Our system performed much better on Subtask A comparing to Subtask B, which reveals that the semi-ranking task is inherently more sophisticated than the pair-wise comparison task. It also suggests that the language models we used are weak when it comes to a more complicated situation. Going forward we should pay extra attention on developing systems that handle Subtask B better.

Last but not the least, although our system performed well for the task, there is evidence showing that neural network models can out perform traditional back-off N-gram language models (Mikolov et al., 2011). In the future we would like to experiment deep learning methods such as Long Short Term Memory (LSTM) neural networks on language models since these networks are capable of forming short term memory and may be better suited for dealing with sequence data. Furthermore, in their work of learning a sense of humor, (Potash et al., 2016) pointed out that external knowledge such as movie names and book tiles is crucial for this task in some cases in order to judge if a tweet is funny responding to a certain hashtag. One innovative way of interacting with external knowledge is to implement a Neural Turing Machine (Graves et al., 2014). In the future we intend to look into approaches of making use of external resources to improve our system performance.

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.

Daniel Jurafsky and James H. Martin. 2009. *Speech and Language Processing (2Nd Edition)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

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

Tomáš Mikolov, Stefan Kombrink, Lukáš Burget, Jan Černockỳ, and Sanjeev Khudanpur. 2011. Extensions of recurrent neural network language model. In *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*. IEEE, pages 5528–5531.

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.

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.