Can We Make Computers Laugh at Talks?

Chong Min Lee
Educational Testing Service
660 Rosedale Road
Princeton, NJ 08541 USA
clee001@ets.org

Su-Youn Yoon
Educational Testing Service
660 Rosedale Road
Princeton, NJ 08541 USA
syoon@ets.org

Lei Chen
Educational Testing Service
660 Rosedale Road
Princeton, NJ 08541 USA
lchen@ets.org

Abstract

Considering the importance of public speaking skills, a system that can predict where audiences might laugh during a talk can be helpful to a person preparing for a presentation. We investigated the possibility that a state-of-the-art humor recognition system could be used to detect sentences that induce laughters. In this study, we used TED talks and audience laughters during those talks as data. Our results showed that the state-of-the-art system needs to be improved in order to be used in a practical application. In addition, our analysis showed that classifying humorous sentences in talks is very challenging due to the close similarity between humorous and non-humorous sentences.

1 Introduction

Public speaking is an important skill for delivering knowledge or opinions to public audiences. In order to develop a successful talk, it is common to practice presentations, with colleagues acting as simulated audiences who then offer their feedback. A recent focus on the importance of public speaking led various studies (Batrinca et al., 2013; Kurihara et al., 2007; Nguyen et al., 2012) to develop systems for automatically evaluating public speaking skills. These studies used audio and video cues in order to evaluate the overall aspects of public speaking. However, the collection of human evaluation data for such systems is time-consuming and challenging (Chen et al., 2014).

If it is shown to be easier to collect audiences' reactions, it may also make sense to explore building an automated system which provides expected audience reactions. For example, speakers sometimes try to add sentences that make audiences laugh or applaud in order to make a successful talk. As Gruner (1985) said, humor in public speakings will "produce a more favorable reaction toward a speaker" and "enhance speaker image." However, there is no guarantee that the expected reactions would occur in an actual talk. If an automatic system can provide audience reactions which are likely to occur in actual talks, it will be helpful in the process of preparing a talk. In this study, we investigated the feasibility of current NLP technologies in building a system which provides expected audience reactions to public speaking.

Studies on automatic humor recognition (Mihalcea and Strapparava, 2005; Yang et al., 2015; Zhang and Liu, 2014; Purandare and Litman, 2006) have defined the recognition task as a binary classification task. So, their classification models categorized a given sentence as a humorous or non-humorous sentence. Among the studies on humor classification, Mihalcea and Strapparava (2005) and Yang et al. (2015) reported high performance on the task. Considering the performance of their systems, it is reasonable to test the applicability of their models to a real application. In this study, we specifically applied a state-of- the-art automatic humor recognition model to talks and investigated if the model could be used to provide simulated laughters.

In our application of the state-of-art system to talks, we could not achieve a comparable performance to the reported performance of the system. We investigated the potential reasons for the performance

Proceedings of the Workshop on Computational Modeling of People's Opinions, Personality, and Emotions in Social Media, pages 173–181, Osaka, Japan, December 12 2016.

difference through further analysis. Some humor classification studies (Mihalcea and Strapparava, 2005; Yang et al., 2015; Barbieri and Saggion, 2014) have used negative instances from different domains or topics, because non-humorous sentences could not be found or are very challenging to collect in target domains or topics. Their studies showed that it was possible to achieve promising performance using data from heterogeneous domains. However, our study showed that humorous sentences which were semantically close to non-humorous sentences were very challenging to distinguish.

We first describe previous studies related to our study. Then, the data we used is described. The descriptions of our experiments and results follow. Our first experiment was to apply a state-of-the-art humor classification system to talks. We then conducted additional experiments and analysis in order to see the impact of domain differences on humor classification tasks.

2 Background

Previous studies (Mihalcea and Strapparava, 2005; Yang et al., 2015; Zhang and Liu, 2014; Purandare and Litman, 2006; Bertero and Fung, 2016) dealt with the humor recognition task as a binary classification task, which was to categorize a given text as humorous or non-humorous. These studies collected textual data which consisted of humorous texts and non-humorous texts and built a classification model using textual features. Humorous and non-humorous texts were from different domains across the studies. Pun websites, daily joke websites, or tweets were used as sources of humorous texts. Resources such as news websites, proverb websites, etc. were used as sources of non-humorous texts. Yang et al. (2015) tried to minimize genre differences between humorous and non-humorous texts in order to avoid a chance that a trained model was optimized to distinguish genre differences. Barbieri and Saggion (2014) examined cross-domain application of humor detection systems using Twitter data. For example, they trained a model using tweets with '#humor' and '#education' hashtags and evaluated the performance of the model on evaluation data containing tweets with '#humor' and '#politics' hashtags. They also reported promising performance in the cross-domain application. These studies which used data from different domains or topics reported very high performance – around 80% accuracy.

Distinct from the other studies, Purandare and Litman (2006) used data from a single domain, the famous TV series, *Friends*. In their study, the target task was to categorize a speaker's turn as humorous or non-humorous. Speakers' turns which occurred right before simulated laughters were defined as humorous ones and the other turns as non-humorous ones. Another difference from other studies was that their study used speakers' acoustic characteristics as features. Their study reported low performance of around 0.600 accuracy for the classification task. Bertero and Fung (2016) pursued similar hypothesis to Purandare and Litman (2006). In their study, the target task was to categorize an utterance in a sitcom, *The Big Bang Theory*, into those followed by laughters or not. Their study was the first study where a deep learning algorithm was used for humor classification.

In this study, our target task was to categorize sentences in talk data into humorous and non-humorous sentences. We only examined textual features. Compared to previous studies, one innovation of this study was that a trained model was evaluated using humorous and non-humorous sentences from the same genre and same topic. Mihalcea and Strapparava (2005) and Yang et al. (2015) borrowed negative instances from different genres such as news websites or proverbs. Barbieri and Saggion (2014) borrowed negative instances from different topics among tweets, though both their positive and negative instances came from the same genre, tweets. Our talk data, on the other hand, was distinct from Barbieri and Saggion (2014) in that negative instances were selected from the same talks as positive instances. As a result, negative instances were inherently from the same topic as corresponding positive instances. In addition, we used real audience reactions (audience laughters) in building our data set. So, the task of this study was to categorize sentences into sentences which made audiences laugh or not, in a talk.

3 Data and Features

3.1 Pun of Day Data

Yang et al. (2015) collected a corpus of Pun of Day data [1]. The data consisted of 2,423 humorous (positive) texts and 2,403 non-humorous (negative) texts. The humorous texts were from the Pun of the Day website, and the negative texts from AP News2, New York Times, Yahoo! Answers and Proverb websites. Examples of humorous and non-humorous sentences are given below.

Humorous `The one who invented the door knocker got a No-bell prize.`
Non-Humorous `The one who discovered/invented it had the last name of fahrenheit.`

In order to reduce the differences between positive and negative instances in the data, Yang et al. (2015) used two constraints when collecting negative instances. Non-humorous texts were required to have lengths between the minimum and maximum lengths of positive instances, in order to be selected as negative instances. In addition, only non-humorous texts which consisted of words found in positive instances were collected.

3.2 TED Talk Data

TED Talks [2] are recordings from TED conferences, and other special TED programs. Corresponding transcripts of most TED Talks are available online. We used the transcripts of the talks as data. Most transcripts of the talks contain the markup '(Laughter)', which represents where audiences laughed aloud during the talks. In addition, time stamps are available in the transcripts. An example transcription is given below [3].

1:14 `...My mother said that she thought I'd really rather have a blue balloon. But I said that I definitely wanted the pink one. And she reminded me that my favorite color was blue. The fact that my favorite color now is blue, but I'm still gay -- (Laughter) -- is evidence of both my mother's influence and its limits.`
1:57 `(Laughter)`
2:06 `When I was little, my mother used to say, ...`

After collecting TED Talk transcripts [4], we manually cleaned up the data. First, we removed transcripts of talks which contained performance like dance or music (e.g. `http://www.ted.com/talks/a_choir_as_big_as_the_internet`). Then, transcripts without '(Laughter)' markups were removed. Other transcripts which we excluded were talks in languages other than English. After the cleaning, the final remaining data set contained 1,192 transcripts.

Following the manual cleaning, we split the transcripts into sentences using the Stanford CoreNLP tool (Manning et al., 2014), then categorized the sentences into humorous and non-humorous sentences. Humorous sentences were sentences which contained or were immediately followed by '(Laughter)'. The other sentences were categorized as non-humorous sentences. The numbers of humorous and non-humorous sentences were 5,801 (3%) and 168,974 (97%), respectively.

When giving a talk, a speaker can induce laughters using means other than language, such as silly gestures. For example, audiences laughed after the sentence 'But, check this out.' in a TED Talk video because the speaker showed a funny picture. We tried to include only humorous sentences where the language alone induced laughters, because we only used textual features. In selecting humorous sentences, we used a simple heuristic. When laughters occurred after a very short sentence which consisted of fewer than seven words, it was likely that the laughters were due to something other than the sentence itself.

[1] The authors of Yang et al. (2015) kindly shared their data with us. We would like to thank them for their generosity.
[2] `http://www.ted.com`
[3] `https://www.ted.com/talks/andrew_solomon_love_no_matter_what/transcript?language=en\#t-284230`
[4] Transcripts were collected on 7/9/2015.

'Pun of the Day' data can provide indirect support for our threshold because the humorous content of 'Pun of the Day' data is solely textual. The average length of 'Pun of the Day' data was 14 words, with a standard deviation of 5. The number of humorous sentences left after removing sentences with fewer than seven words was 4,726.

Utilizing the same experimental setup as Mihalcea and Strapparava (2005) and Yang et al. (2015) (50% positive and 50% negative instances), we selected 4,726 sentences from among all collected non-humorous sentences as negative instances. During selection, we minimized differences between positive and negative instances. A negative instance was selected from among sentences located close to a positive instance in a talk. We made a candidate set of non-humorous sentences using sentences within a window size of seven (e.g. from `sent-7` to `sent-1` and from `sent+1` to `sent+7` in the following):

sent-7 ...
...
sent-1 `And she reminded me that my favorite color was blue.`
Humorous `The fact that my favorite color now is blue, but I'm still gay is evidence of both my mother's influence and its limits.`
sent+1 `When I was little, my mother used to say, ...`
...
sent+7 ...

Among the candidates, sentences which consisted of less than seven words were removed and a negative instance was randomly selected among the remaining ones.

3.3 Implementation of Features

Features from Yang et al. (2015), which we implemented, consisted of (1) two incongruity features, (2) six ambiguity features, (3) four interpersonal effect features, (4) four phonetic features, (5) five *k*-Nearest Neighbor features, and (6) 300 Word2Vec features. The total number of features used in this study was 321. We describe our implementation of the features in this section. The justifications for the features can be found in the original paper.

Incongruity Features: the existence of incongruous or incompatible words in a text can cause laughters (e.g. *A **clean** desk is a sign of a **cluttered** desk drawer.* (Mihalcea and Strapparava, 2005)). We calculated meaning distances of all word pairs in a sentence using a Word2Vec implementation in Python [5]. The maximum and minimum meaning distances among the calculated distances in a sentence were used as two incongruity features.

Ambiguity Features: the use of ambiguous words in a sentence can also trigger humorous effects (i.e. *A political prisoner is one who stands behind her **convictions***. (Miller and Gurevych, 2015)). We calculated sense combinations of nouns, verbs, adjectives and adverbs. We made four groups, composed of the nouns, verbs, adjectives and adverbs in a sentence, respectively. Then, we collected counts of possible meanings of each word in each group from WordNet (Fellbaum, 1998). For example, when two nouns in a sentence have two and three different meanings in WordNet, the sense combination of the noun group was 1.792 ($log(2 \times 3)$). We also calculated the largest and smallest WordNet Path Similarity values of pairs of words in a sentence using a Python interface for WordNet [6].

Interpersonal Effect Features: sentences can be humorous when sentences contain strong sentiment or subjectivity words (Zhang and Liu, 2014). In TED Talk data, some positive instances also contain strong sentiment words (i.e. *Then, just staying above the Earth for one more second, people are acting like **idiots** all across the country.*) We extracted the number of occurrences of all negative (positive) polarity words and the number of weak (strong) subjectivity words using the word association resource from Wilson et al. (2005).

Phonetic Style: phonetic properties such as alliteration and rhyme can make people laugh (i.e. ***Infan**ts don't enjoy **infan**cy like **adul**ts do **adult**ery.* (Mihalcea and Strapparava, 2005)) Using the CMU

[5] `https://radimrehurek.com/gensim/models/word2vec.html`
[6] `http://www.nltk.org/howto/wordnet.html`

	Accuracy	F1 Score	Precision	Recall
Yang	85.4%	85.9%	83.4%	88.8%
Pun-to-Pun	85.7%	86.4%	82.5%	90.8%
Pun-to-Talk	50.5%	50.1%	50.5%	49.7%
Talk-to-Talk	53.5%	60.3%	52.5%	70.8%
Talk-to-Pun	52.6%	58.5%	52.2%	66.6%

Table 1: The Performances of state-of-the-art system

Pronuncing Dictionary, we extracted the number of alliteration chains in a sentence, the maximum length of alliteration chains, the number of rhyme chains, and the maximum length of rhyme chains.

k-Nearest Neighbors Features: We used unigram feature vectors with a k-nearest neighbor algorithm in calculating these features. When a sentence is given, we retrieved labels of the five nearest neighbors in a k-nearest neighbor model using euclidean distance. The five labels were used as features.

Word2Vec Features: we collected Word2Vec embeddings of words in a sentence, then used the average of the embeddings as a representation of the sentence. We used the Google Word2Vec model [7] and the Gensim Python package (Řehůřek and Sojka, 2010).

4 Experiments

4.1 Application of State-of-Art Technology to Talk Data

In this section, we present expeirments that we ran to determine 1) how effective a model trained using 'Pun of Day' data (Pun) is when applied to TED Talk data (Talk), and 2) whether the performance of a model trained using Talk data would be similar to the performance reported in Yang et al. (2015). We reimplemented features developed by Yang et al. (2015) and evaluated those features on Talk data. Considering the different characteristics of Talk data versus Pun data, we sought to investigate whether Yang's model could achieve the reported performance (over 85% accuracy) on our Talk data. The differences were 1) humorous sentences in Talk data were sentences which induced audience laughters, compared to Pun data which used canned textual humor, 2) all non-humorous sentences in Talk data were also from TED talks, and 3) each pair of humorous and non-humorous sentences were semantically close because they were closely placed. These differences made the humor classification task more challenging.

We first validated the performance of the reimplemented features. We followed the experimental setup of Yang et al. (2015) in order to see if the performance of our duplicated features was comparable to their reported performance. Their best performance was 85.4% accuracy (Yang in Table 1) when they used Random Forest as a classifier and 10-fold cross validation (CV) as an evaluation method. Replicating this experiment setup, we were able to achieve 86.0% accuracy (Pun-to-Pun in Table 1), which is slightly better than the performance reported in their paper. The performance difference could be due to the difference in partitions in CV.

After verifying the feature implementation, we built a humor recognition model using the entirety of the Pun data. The model was evaluated on Talk data in order to see how effective a state-of-art model was in spite of differences between the two data sets. The accuracy was only 50.5% (Pun- to-Talk in Table 1) which is 0.5% higher than a majority class classifier. The poor performance observed in this second experiment could be due to the differences between Pun and Talk data. Based on these experimental results, it can be said that a humor classification model trained using Pun data can't be directly used in categorizing humor sentences from talks.

The third experiment was designed to observe the performance of a model (Talk-to-Talk) built using Talk data. The Talk-to-Talk model was evaluated on Talk data using 10-fold CV. When we split Talk data into train and test data in a CV fold, sources of sentences were used as a criteria in the split. All humorous and non-humorous sentences from one talk only belonged to a train data or a test data, not

[7] https://drive.google.com/file/d/0B7XkCwpI5KDYNlNUTTlSS21pQmM/edit?usp=sharing

	Accuracy	F1	Precision	Recall
Talk Pos + Pun Neg	83.9%	84.3%	82.7%	85.8%
Pun Pos + Talk Neg	82.6%	83.4%	79.8%	87.4%

Table 2: The Performance using combined data. 'Pos' and 'Neg' mean 'Positives' and 'Negatives'.

both. This criterion was adopted because sentences from a talk could share contexts and the shared contexts could boost performance. Using the model, we got 53.2% accuracy (Talk-to-Talk in Table 1). Thus, we observed a 3% increase in accuracy and 10% increase in F1 score, when compared with the Pun-to-Talk model. But, the performance was still poorer than Yang's reported performance. The model trained on Talk data showed a preference for categorizing instances in evaluation data into humorous instances, according to the precision and recall values of Talk-to-Talk.

4.2 Cross Domain Data Combinations

In the experiments described in the preceding section, we weren't able to get results comparable to Yang et al. (2015) when Talk data was used in both train and evaluation data. The results of our experiments raised questions about why two different results were observed for two different data sets. A major difference in the two data sets was the source of negative instances. Yang et al. (2015) borrowed negative instances from different genres such as news websites and proverbs. But, in Talk-to-Talk, both positive and negative instances were from the same genre. Furthermore, each humorous instance had a corresponding non-humorous instance from the same talk. In this section, we investigate the impact of genre differences in the humor classification task, using Pun and Talk data.

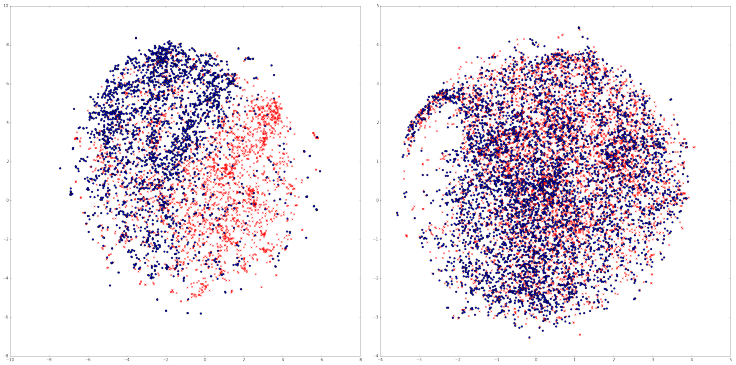
The positive instances (humorous sentences) in the Talk data may be substantially different from the ones found in Pun [8]. Humorous sentences in the Pun data set are 'self-contained'. It means that the point of humor can be understood within a single sentence. On the other hand, the humorous sentences in the Talk data set may be 'discourse-based', which means that the source of humor in target sentences might be understood in the wider context of the speaker's performance. In addition, negative instances of Talk data may also be 'discourse-based', which means that the wider context can be required to understand the sentences. However, the negatives in the Pun data are not 'discourse-based'. It is worth investigating whether the 'discourse-based' characteristics of the Talk data made it impossible to achieve high performance. So, we combined 'discourse-based' instances with 'self-contained' instances and checked if we could achieve high performance using the combined data.

We built two data sets combining positives of Talk and negatives of Pun ('Talk Pos + Pun Neg'), and positives of Pun and negatives of Talk ('Pun Pos + Talk Neg') in order to make data sets containing positives and negatives from different genres. When we trained and evaluated 'Talk Pos + Pun Neg' and 'Pun Pos + Talk Neg' models using 10-fold CV, we could achieve 82.5% and 83.6% accuracies which were similar to Pun-to-Pun performance as observed in Table 2. In both cases of 'Pun Pos + Talk Neg' and 'Talk Pos + Pun Neg', we didn't observe a significant drop in performance. We assumed that 'discourse-based' characteristics of Talk data were difficult to learn based on the low performance of 'Talk-to-Talk' in Table 1. When we looked through humorous instances of Talk data, we observed 'discourse-based' humorous cases which could be difficult to capture using Yang's features (i.e. "this was the worst month of my life", "and I said well that would be great", and "so I wanted to follow that rule"). Of particular interest, we still observed precision and recall as high as 82.7% and 85.8%, respectively. The high performance without a significant drop was counter-intuitive. This observation raised the question of what exactly classifiers learned using the data.

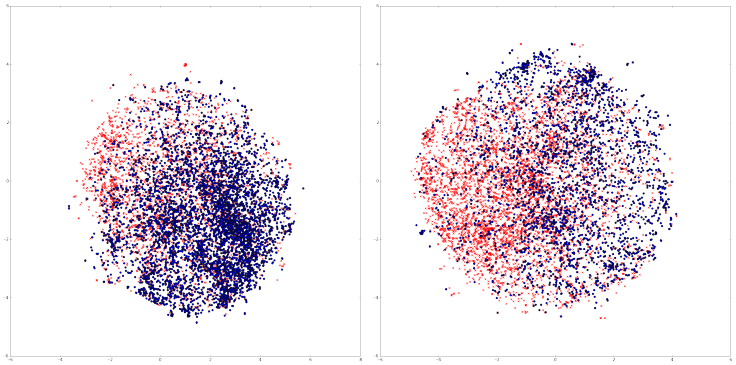
5 Discussion

Through our experiments, we observed higher performances when genre difference existed between positive and negative instances. In contrast, lower performance was achieved without the difference.

[8] We appreciate input from an anonymous reviewer from EMNLP 2016 who pointed out the difference between data.



(a) Pun Positives and Pun Negatives (b) Talk Positives and Talk Negatives



(c) Pun Positives and Talk Negatives (d) Talk Positives and Pun Negatives

Figure 1: Word2Vec feature distribution using t-SNE. In each figure, each blue 'o' or red '+' means a positive or negative instance, respectively

Our hypothesis of the cause of the phenomena was semantic distance between positive and negative data points. Negative instances from Talk data were selected from among sentences within seven preceding and following sentences of positive instances. So, the meaning of a negative instance would be close to the meaning of a corresponding positive instance. But, the meaning of Pun positives would be quite different from the meaning of Pun negatives because they were from different genres although words in positives and negatives of Pun were shared.

Recently, Li et al. (2016) and Arras et al. (2016) showed that it is possible to understand predictions of NLP models by visualizing word embeddings. Following those studies, we also tried to get a hint at the accuracy of our hypothesis through visualizing the Word2Vec embedding features that we used in our experiments. We used the average of Word2Vec embeddings of words in a sentence as a representation of the sentence. We visualized sentence representations using t-SNE (van der Maaten and Hinton, 2008).

As shown in Figure 1a, meanings of Pun positives and negatives were grouped in distinct areas. Pun positives and negatives were positioned at the right bottom area and left upper area, respectively. The combination of Talk positives and Pun negatives was another case containing clearer meaning distinction between positive and negative instances. In the case of the combination of Pun positives and Talk negatives, the distinction was weaker but one can still identify a small group of negatives at the upper left and the somewhat more dispersed group of positives at the bottom right. However, Talk positives and negatives were completely mixed throughout. So, it was impossible to make distinctions on groups of positives and negatives.

This analysis provided clues to the high performances of 'Pun-to-Pun' in Table 1, and 'Talk Pos + Pun Neg' and 'Pun Pos + Talk Neg' in Table 2, as well as the low performance of 'Talk-to-Talk' in Table 1. The high-performance data were much more learnable than 'Talk-to-Talk', based on the above observations about the discreteness of each data set's tokens.

Another analysis we conducted was the impact of the closeness of negatives in Talk data. We selected a negative instance within seven preceding and following sentences of a positive instance. Positive in-

stances of Talk data could be punchlines which brougt up audiences' laughters after laughable mood was built up through preceding sentences. In other words, preceding sentences could be also humorous but not humorous enough to cause laughters. When slightly humorous sentences are included in negative instances, the poor performance of 'Talk-to- Talk' is reasonable because it is very challenging to distinguish humorous sentences from less humorous sentences, even for humans. So, we conducted another experiment after randomly choosing a negative instance among all sentences, which didn't cause laughters, within a talk of a positive instance. Then, we trained and evaluated models using 10-fold CV. In this experiment, we could get 55.4% accuracy which was only 2% higher than 'Talk-to- Talk' in Table 1. This further analysis is a supporting evidence that humor detection in a talk is a challenging task irrespective of the distance in text between positive and negative instances.

6 Conclusions

In this study, we investigated whether a state-of-the-art humor recognition model could be used in simulating audience laughters in talks. Our results showed that lots of improvements in the humor recognition task would be needed in order to be used in real applications. In addition, we showed through the visualization of the features that Talk data is much more difficult for a machine to learn due to the featural closeness of positive and negative instances. We have a plan to develop features on the discouse level, in order to improve the performance. Humorous sentences in TED talks are parts of talks. Preceding sentences before humorous sentences construct contexts. The combination of contents of humorous sentences and established contexts can lead to laughter. We will investigate this conceptual possibility in future work.

References

Leila Arras, Franziska Horn, Grégoire Montavon, Klaus-Robert Müller, and Wojciech Samek. 2016. Explaining predictions of non-linear classifiers in NLP. In *the 1st Workshop on Representation Learning for NLP*, pages 1–7, Berlin, Germany, August. Association for Computational Linguistics.

Francesco Barbieri and Horacio Saggion. 2014. Automatic detection of irony and humour in Twitter. In *Proceedings of the Fifth International Conference on Computational Creativity*, Ljubljana, Slovenia, jun. Josef Stefan Institute, Ljubljana, Slovenia, Josef Stefan Institute, Ljubljana, Slovenia.

Ligia Batrinca, Giota Stratou, Ari Shapiro, Louis-Philippe Morency, and Stefan Scherer. 2013. Cicero - Towards a Multimodal Virtual Audience Platform for Public Speaking Training. In *International Conference on Intelligent Virtual Humans*, Lecture Notes on Computer Science, pages 116–128, Edinburgh, UK, August.

Dario Bertero and Pascale Fung. 2016. A long short-term memory framework for predicting humor in dialogues. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 130–135, San Diego, California, June. Association for Computational Linguistics.

Lei Chen, Gary Feng, Jilliam Joe, Chee Wee Leong, Christopher Kitchen, and Chong Min Lee. 2014. Towards automated assessment of public speaking skills using multimodal cues. In *Proceedings of the 16th International Conference on Multimodal Interaction*, ICMI '14, pages 200–203, New York, NY, USA. ACM.

Christiane Fellbaum. 1998. *WordNet: An Electronic Lexical Database*. Bradford Books.

Charles R. Gruner. 1985. Advice to the beginning speaker on using humorwhat the research tells us. *Communication Education*, 34(2):142–147.

Kazutaka Kurihara, Masataka Goto, Jun Ogata, Yosuke Matsusaka, and Takeo Igarashi. 2007. Presentation sensei: A presentation training system using speech and image processing. In *Proceedings of the 9th International Conference on Multimodal Interfaces*, ICMI '07, pages 358–365, New York, NY, USA. ACM.

Jiwei Li, Xinlei Chen, Eduard H. Hovy, and Dan Jurafsky. 2016. Visualizing and understanding neural models in NLP. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pages 681–691.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60.

Rada Mihalcea and Carlo Strapparava. 2005. Making computers laugh: Investigations in automatic humor recognition. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 531–538, Vancouver, British Columbia, Canada, October. Association for Computational Linguistics.

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)*, pages 719–729, Beijing, China, July. Association for Computational Linguistics.

Anh-Tuan Nguyen, Wei Chen, and Matthias Rauterberg. 2012. Online feedback system for public speakers. In *E-Learning, E-Management and E-Services (IS3e), 2012 IEEE Symposium on*, pages 1–5, Oct.

Amruta Purandare and Diane J. Litman. 2006. Humor: Prosody analysis and automatic recognition for F*R*I*E*N*D*S*. In *EMNLP*.

Radim Řehůřek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, May. ELRA. `http://is.muni.cz/publication/884893/en`.

Laurens van der Maaten and Geoffrey E. Hinton. 2008. Visualizing high-dimensional data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605.

Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, pages 347–354, Stroudsburg, PA, USA. Association for Computational Linguistics.

Diyi Yang, Alon Lavie, Chris Dyer, and Eduard Hovy. 2015. Humor recognition and humor anchor extraction. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2367–2376, Lisbon, Portugal, September. Association for Computational Linguistics.

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*, pages 889–898, New York, NY, USA. ACM.