Deep Learning of Audio and Language Features for Humor Prediction

Dario Bertero, Pascale Fung
Human Language Technology Center
Department of Electronic and Computer Engineering
The Hong Kong University of Science and Technology, Clear Water Bay, Hong Kong
dbertero@connect.ust.hk, pascale@ece.ust.hk

Abstract

We propose a comparison between various supervised machine learning methods to predict and detect humor in dialogues. We retrieve our humorous dialogues from a very popular TV sitcom: "The Big Bang Theory". We build a corpus where punchlines are annotated using the canned laughter embedded in the audio track. Our comparative study involves a linear-chain Conditional Random Field over a Recurrent Neural Network and a Convolutional Neural Network. Using a combination of word-level and audio frame-level features, the CNN outperforms the other methods, obtaining the best F-score of 68.5% over 66.5% by CRF and 52.9% by RNN. Our work is a starting point to developing more effective machine learning and neural network models on the humor prediction task, as well as developing machines capable in understanding humor in general.

Keywords: humor prediction, neural networks, TV-sitcoms

1. Introduction

The term "humor" refers to various kinds of stimuli, including acoustic, verbal, visual and situational, that are able to trigger a laughter reaction in the recipient. It is an important aspect of our everyday life, and is supposed to give benefits to physical and psychological health (Sumners, 1988; Martineau, 1972; La Fave et al., 1976; Anderson and Arnoult, 1989; Lefcourt et al., 1997; Lefcourt and Martin, 2012).

There has recently been many attempts in detecting humor from canned jokes (Yang et al., 2015), customer reviews (Reyes and Rosso, 2012) and Twitter (Reyes et al., 2013; Barbieri and Saggion, 2014; Riloff et al., 2013; Joshi et al., 2015). All these analyses are only on isolated textual data. Fewer work took into consideration other elements, such as the surrounding context (Bamman and Smith, 2015; Karoui et al., 2015) or acoustic and prosodic features (Rakov and Rosenberg, 2013).

We propose to predict when people would laugh in a dialog with a supervised machine learning approach. While most of the past attempts concentrate on isolated examples, the response to humor in a conversation depends heavily on the surrounding context, such as the conversational topic and the previous utterances. It is quite common that the same utterance may trigger a different effect on the recipient depending on when it is used. Two distinct moments can be identified in humor and joke generation: a setup where appropriate inputs are given and the context for the joke is built, and the "punchline" where the climax is reached and people are triggered to react with laugh (Hetzron, 1991; Attardo, 1997). Our task is to identify these punchlines and thus predict where laughter occurs in the dialog flow. Moreover the way a spoken dialog utterance is made is another important element that may trigger a humorous reaction. Thus we also propose to combine acoustic and language features.

To meet our objectives we build a corpus with dialogues taken from a popular TV sitcom: "The Big Bang Theory". TV sitcoms are a good source of both acoustic speech data from the audio tracks, and their transcriptions from the subtitle files. They are embedded with canned laughter which provide pretty good indication of when in the show the audience is expected to laugh.

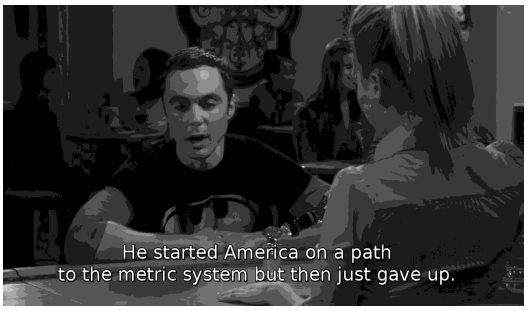

Figure 1: PENNY: *Okay, Sheldon, what can I get you?*
SHELDON: *Alcohol.*
PENNY: *Could you be a little more specific?*
SHELDON: ***Ethyl alcohol.*** **LAUGH** ***Forty milliliters.*** **LAUGH**
PENNY: *I'm sorry, honey, I don't know milliliters.*
SHELDON: ***Ah. Blame President James Jimmy Carter.*** **LAUGH** ***He started America on a path to the metric system but then just gave up.*** **LAUGH**

An example of dialog from this sitcom is shown in Figure 1. Before each punchline, in bold, are the utterances which build the setup for the joke. It is quite evident that some punchlines might not trigger any reaction, or are much less effective, without the proper context (such as the fact the conversation is held in a bar) and the proper setup.

In order to fully take advantage of the dialog context, we employ and compare three different classification algorithms: a Conditional Random Field, a Recurrent Neural Network and a Convolutional Neural Network. We train the former two with a set of acoustic and language features, while in the latter we replace some of the features with low level representations of words and acoustic frames.

Predicting when people would react to humor and laugh is an important problem with potential great implications in

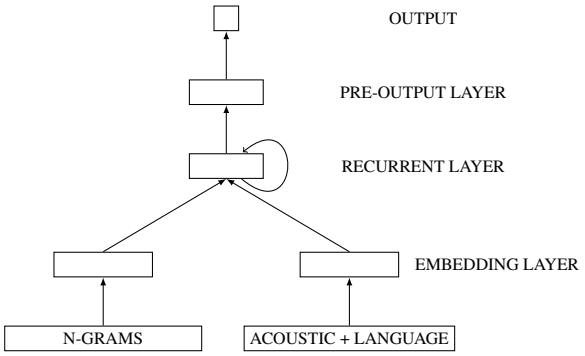

Figure 2: RNN structure.

human-machine interaction. A system that predict humor is a foundational block for future empathetic machines able to effectively understand and react to humorous stimuli provided by the user (Fung, 2015).

2. Methodology

We propose a supervised classification approach based on the combined contribution of acoustic and language features. Furthermore we are interested in comparing the performances of different classifiers such as a Conditional Random Field (Lafferty et al., 2001; Bertero and Fung, 2016), a Recurrent Neural Network (Elman, 1990) and a Convolutional Neural Network (Collobert et al., 2011). We also train a simple Logistic Regression baseline.

2.1. Acoustic features

In a multimodal dialog variations in pitch, loudness and intonation often indicate whether the intent of the speaker is serious or humorous. To model this aspect we retrieve a set of around 2500 acoustic features from the openSMILE software (Eyben et al., 2013) using the emobase and emobase2010 packages provided (made of the feature set from the INTERSPEECH 2010 paralinguistic challenge (Schuller et al., 2010)). These features include MFCC, pitch, intensity, loudness, probability of voicing, F_0 envelope, Line Spectral Frequencies, Zero-Crossing rate and their variations (delta coefficients).
Another element that is associated with humor is the speed at which an utterance is said. Talking deliberately too slowly may make fun of the recipient, while a deliberate fast pace may prevent the listener to catch all the information and trigger violation of Gricean Maxim of manner (Attardo, 1993). We therefore include the speaking rate of the utterance (time duration divided by the number of words) to our feature set.

2.2. Language features

We also retrieve a set of language features from the utterance transcriptions. They represent multiple aspects, ranging from syntax to semantic and sentiment. The features we use are:

- Lexical: unigrams, bigrams ans trigrams that appear 5 times or more.
- Syntactic and structural (Barbieri and Saggion, 2014): proportion of nouns, verbs, adjectives and adverbs, sentence length, length difference with the previous utterance and average word length.
- Sentiment (Barbieri and Saggion, 2014): average of positive sentiment scores and negative sentiment scores from SentiWordNet, average of all scores and difference between the positive and negative averages.
- Antonyms: presence of noun, verb, adjective and adverb antonyms in the previous utterance, obtained from WordNet (Miller, 1995).
- Speaker turns: speaker identity and position within the speaker turn (beginning, middle, end, isolated). Various speakers are more or less likely to generate humor (as shown in figure 4).

2.3. Conditional random field (CRF)

The CRF is a popular sequence tagging algorithm for modeling time sequences. It gives good performance when dealing with similar time-variant data, in tasks such as disfluency detection (Liu et al., 2006) and text summarization (Zhang and Fung, 2012). We use a standard linear chain CRF to model our dialog, which can be summarized with the following equation:

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \prod_A \mathbf{exp}\left\{ \sum_k \theta_{Ak} f_{Ak}(\mathbf{x}_A, \mathbf{y}_A) \right\} \quad (1)$$

where A represents the graph nodes, k is the feature index, $\mathbf{x}$ is the total observation, θ_{Ak} are the model parameters to be trained, f_{Ak} are the feature functions and $Z(\mathbf{x})$ a normalization function.

2.4. Recurrent Neural Network (RNN)

The RNN is a neural network layout that provides a memory component to the classifier, in the form of a recurrent layer that is fed back as input at every time instant. It has been used with great success in tasks such as language modeling (Bengio et al., 2003), where the recurrent layer keeps track of the past context in order to effectively predict the following tokens.
A diagram of our network layout is shown in figure 2. The language and acoustic feature sets are first fed into separate embedding layers of the form:

$$\mathbf{x}_t^{emb} = \tanh(\mathbf{W}_{emb}\mathbf{x}_t + \mathbf{b}_{emb}) \quad (2)$$

where $\mathbf{W}$ and $\mathbf{b}$ are the parameters to train. The embedding layer is used to rerange the two feature vector and reduce their dimensionalities, in order to balance their contributions.
The two vectors obtained are concatenated together and given as input to the recurrent layer, which has the form:

$$\mathbf{h}_t = \tanh(\mathbf{W}_h\mathbf{h}_{t-1} + \mathbf{W}_x\mathbf{x} + \mathbf{b}_{rnn}) \quad (3)$$

where $\mathbf{x}$ is the input and $\mathbf{h}_{t-1}$ the hidden layer at the previous time instant. This kind of backpropagation has the ability of retaining information about the past utterances.

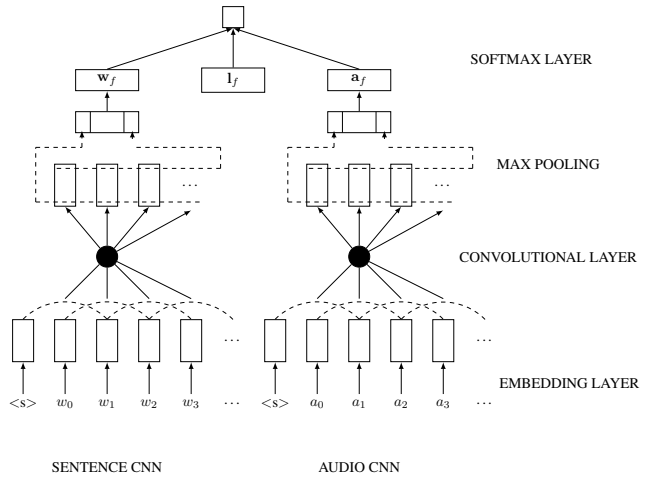

Figure 3: CNN structure. w_i are the Word2Vec input vectors, a_i the audio frames feature input vectors. $\mathbf{w}_f$ is the output of the sentence encoding CNN, $\mathbf{a}_f$ the output of the audio encoding CNN, $\mathbf{l}_f$ the other features vector.

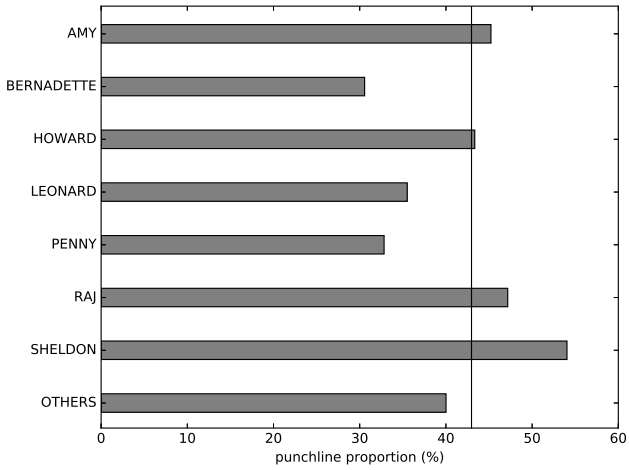

Figure 4: Proportion (percentage) of punchlines for the most frequent characters. The vertical line represents the overall average of 42.8%.

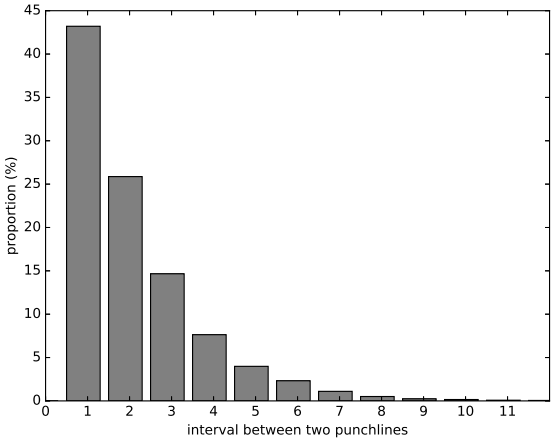

Figure 5: Distribution of intervals between two punchlines. In The Big Bang Theory, on average, it is equal to 2.2 utterances.

We apply another layer before the output softmax layer to enhance the results (Pascanu et al., 2013).
In our specific task the RNN is intended to model the setup-punchline structure of conversational humor. The hidden layer should model the setup of each scene remembering the previous utterances and keeping track of the context that leads to each punchline. It should provide an advantage over simpler classifiers such as logistic regression, as they are only able to deal with each sample in isolation, or eventually with fixed length context windows.

2.5. Convolutional Neural Network (CNN)

The CNN is another kind of neural network useful to encode a linear or multidimensional structure such a sentence or an image into a fixed-length vector. Previous work has shown that neural network model is particularly effective for extracting and selecting features from low-level input representations (Wang and Manning, 2013). We therefore are interested to evaluate whether using a CNN to encode an utterance from word and audio frame-level inputs may yield higher results than using bag-of-ngram representations or utterance-level acoustic features (Wang and Manning, 2013; Han et al., 2014).
Our network diagram is shown in figure 3. We use two different CNNs to replace respectively the n-gram features and the acoustic features (except the speaking rate) of an utterance. Our first CNN takes as input a word vector for each token taken from Word2Vec (Mikolov et al., 2013). For the second CNN instead we divide the audio track of each utterance into overlapping frames of $25ms$, shifted $10ms$ each other. Then we extract from each frame a subset of lower-level acoustic features from openSMILE. The features we use in this stage include MFCC, pitch, energy, zero-crossing mean, Δ and $\Delta\Delta$. Each CNN is made of an embedding layer that reduces the dimensionality of each input vector. A second layer performs the convolution over a sliding window of 5 tokens for the text case, and 3 frames for the audio network. A max-pooling operation is then applied to reduce all the vectors obtained from the convolution into a single one, selecting the most salient features. A last layer is used to rerange the vector obtained from the max-pooling. To perform the final classification for each utterance we concatenate the outputs from the two CNNs together with the other features (speaking rate and other language features).

3. Experiments

3.1. Corpus

We built a corpus from "The Big Bang Theory" seasons 1 to 6, a very popular humorous TV sitcom. We retrieved the audio tracks, the subtitle files associated, and the scripts (from https://bigbangtrans.wordpress.com). Subtitle files provide the timestamps used to cut the audio tracks into the individual utterances, while the script files include information about the character who speak each utterance and the speaker turns, as well as the division of the episode into scenes.

Classifier and features	Accuracy	Precision	Recall	F-score
All positive baseline	42.8	42.8	100.0	59.9
All negative baseline	57.2	0.0	0.0	0.0
Logistic regression: n-grams	57.6	50.5	47.9	49.2
Logistic regression: acoustic + language	72.0	70.3	59.9	64.7
Logistic regression: all features	72.1	69.1	62.9	65.9
CRF n-grams	61.8	56.8	45.1	50.2
CRF acoustic + language	73.4	**72.1**	61.8	66.5
CRF n-grams + acoustic + language	71.3	68.3	61.3	64.7
RNN n-grams	61.3	57.5	36.5	44.7
RNN acoustic + language	61.3	56.5	41.1	47.6
RNN n-grams + acoustic + language	65.8	64.4	44.9	52.9
CNN lexical	63.8	63.7	35.9	46.0
CNN acoustic only	64.2	59.0	53.6	56.2
CNN lexical + acoustic + language	**73.8**	70.3	**66.7**	**68.5**

Table 1: Results, percentage

To annotate the punchline utterances we retrieved the canned laughters timestamps from the audio track using a vocal removal tool followed by a silence/sound detector tool. The vocal removal tool removes all the voice and gives as output an audio track consisting only of canned laughters, whose time intervals are easily detected by the sound detector. Then we compared the position of the laughters with the utterance timestamps obtained from the subtitles, labeling each utterance immediately or within $1s$ followed by a laughter as a punchline. We also used the canned laughter timing information to cut the laughter from the utterances audio tracks, in order to avoid an eventual bias of the classifier. Moreover we divided each episode into scenes and each utterance with the speaking character, according to the script files.

Overall the corpus contains 1589 scenes. The episodes were divided into a training set of around 35865 overall utterances, and a development set of 3904 and test set of 3903. The corpus consist of 42.8% of the utterances being punchlines. The average interval between two of them is 2.2 utterances, figure 5 shows the overall interval distribution. There are 7 recurring characters appearing for more than 500 utterances. As shown in figure 4 the amount of punchlines associated to each character is different by over 20%. We grouped all characters other than the seven most frequent into the "other" label for the speaker identity feature.

3.2. Experimental setup

In the CRF experiments we used the CRFsuite implementation (Okazaki, 2007) with L2 regularization. In the RNN all the embedding and hidden layers were set to a dimension of 100, and the sigmoid function was chosen as non-linearity, as it gave better performance than the hyperbolic tangent. We trained the network using standard backpropagation with L2 regularization. In the CNN case instead we fix the dimension to 100 for the language CNN and 50 for the acoustic CNN. We obtained the best performance using the hyperbolic tangent non-linearity function in the language CNN, and rectified linear units in the audio CNN. All neural networks were implemented using THEANO toolkit (Bergstra et al., 2010).

Both in the CRF and in the RNN we fed each scene as a separate unit, and in the RNN we reset the recurrent layer after the end of each scene. We used the development set to tune the hyperparameters, and in the case of the neural networks to determine the early stopping condition when the results on it began to get lower.

We made three kinds of experiments with different features: the first one with only the sparse bag-of-ngrams, the second with a set of acoustic and language features excluding n-grams, and the third one combining all the features. For each utterance, with the exception of acoustic features, we use a context window of size 3 including the utterance and the two previous ones. We compare our results with an all positive/all negative baseline, and with a logistic regression classifier trained on the same feature sets. In the CNN case, we evaluated separately the performance in dealing with lexical features only, and with acoustic features only, and we then combined them together adding the other features. Results are shown in table 1.

3.3. Results and discussion

Our results show that the CNN achieves the best overall performance with an F-score of 68.5%, 2% more than the best result obtained from the CRF. The CRF is still quite effective, and it reaches the best overall precision of 72.1% when trained without the bag-of-ngram features. The CRF is slightly better than using a simple logistic regression, as it is able better exploit the sequential structure of the data. This is due in particular to the fact it models the different transition probabilities between setup and punchlines.

From the results obtained it seems that the main advantage of the CNN over the CRF is when dealing with lexical and acoustic features. The convolution applied by the CNN over words and audio-frames is more effective in encoding a sentence than simpler bag-of-ngram representations or high-level acoustic features extracted from the whole utterance. This is particularly evident when the two CNNs are jointly trained. The CNN instead does not model the dialog past context, and this is clear from the results obtained from lexical features only.

The RNN in theory should have been the most suited algorithm to capture the conversational humor structure. We

were expecting an higher performance than the CRF, but the results obtained are instead much lower than all the baselines. The RNN is in general a difficult algorithm to train effectively and is prone to overfitting easily the training data (Pascanu et al., 2012), and it generally need more data to be effectively trained. The input features may also not have been the most suited for this classifier.

To conclude our discussion, it is worth noting that canned laughter are a good indication of laughter response, but it is not perfect. They are primarily intended to solicit regular laughter response in the audience to keep a constant amusement level in the show, and often used to enhance weak jokes.

4. Conclusion

We carried out a comparative study on different supervised machine learning algorithms to predict when people would laugh in a funny dialog. We achieved the best result of 73.8% accuracy with a CNN based framework which encodes and merges together word-level and acoustic-frame level features.

We plan in the future to improve the dialog context modeling, in particular for the CNN case. We are interested in trying other different network structures, such as to replace the RNN with a Long Short-Term Memory, and using it after the CNN output to incorporate the dialog context. Our ultimate goal is to integrate laughter response prediction in a machine dialog system, to allow it to understand and react to humor.

5. Acknowledgments

This work was partially funded by the Hong Kong Phd Fellowship Scheme, and partially by grant #16214415 of the Hong Kong Research Grants Council.

6. Bibliographical References

Anderson, C. A. and Arnoult, L. H. (1989). An examination of perceived control, humor, irrational beliefs, and positive stress as moderators of the relation between negative stress and health. *Basic and Applied Social Psychology*, 10(2):101–117.

Attardo, S. (1993). Violation of conversational maxims and cooperation: The case of jokes. *Journal of pragmatics*, 19(6):537–558.

Attardo, S. (1997). The semantic foundations of cognitive theories of humor. *Humor-International Journal of Humor Research*, (10):395–420.

Bamman, D. and Smith, N. A. (2015). Contextualized sarcasm detection on twitter. In *Ninth International AAAI Conference on Web and Social Media*.

Barbieri, F. and Saggion, H. (2014). Modelling irony in twitter: Feature analysis and evaluation. In *Proceedings of Language Resources and Evaluation Conference (LREC)*, pages 4258–4264.

Bengio, Y., Ducharme, R., Vincent, P., and Janvin, C. (2003). A neural probabilistic language model. *The Journal of Machine Learning Research*, 3:1137–1155.

Bergstra, J., Breuleux, O., Bastien, F., Lamblin, P., Pascanu, R., Desjardins, G., Turian, J., Warde-Farley, D., and Bengio, Y. (2010). Theano: a CPU and GPU math expression compiler. In *Proceedings of the Python for Scientific Computing Conference (SciPy)*, June. Oral Presentation.

Bertero, D. and Fung, P. (2016). Predicting humor response in dialogues from tv sitcoms. In *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE.

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., and Kuksa, P. (2011). Natural language processing (almost) from scratch. *The Journal of Machine Learning Research*, 12:2493–2537.

Elman, J. L. (1990). Finding structure in time. *Cognitive science*, 14(2):179–211.

Eyben, F., Weninger, F., Gross, F., and Schuller, B. (2013). Recent developments in opensmile, the munich open-source multimedia feature extractor. In *Proceedings of the 21st ACM International Conference on Multimedia*, MM '13, pages 835–838, New York, NY, USA. ACM.

Fung, P. (2015). Robots with heart. *Scientific American*, 313(5):60–63.

Han, K., Yu, D., and Tashev, I. (2014). Speech emotion recognition using deep neural network and extreme learning machine. In *Interspeech*, pages 223–227.

Hetzron, R. (1991). On the structure of punchlines. *HUMOR: International Journal of Humor Research*.

Joshi, A., Sharma, V., and Bhattacharyya, P. (2015). Harnessing context incongruity for sarcasm detection. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, volume 2, pages 757–762.

Karoui, J., Farah, B., Moriceau, V., Aussenac-Gilles, N., and Hadrich-Belguith, L. (2015). Towards a contextual pragmatic model to detect irony in tweets. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 644–650, Beijing, China, July. Association for Computational Linguistics.

La Fave, L., Haddad, J., and Maesen, W. A. (1976). Superiority, enhanced self-esteem, and perceived incongruity humour theory.

Lafferty, J., McCallum, A., and Pereira, F. C. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

Lefcourt, H. M. and Martin, R. A. (2012). *Humor and life stress: Antidote to adversity*. Springer Science & Business Media.

Lefcourt, H. M., Davidson, K., Prkachin, K. M., and Mills, D. E. (1997). Humor as a stress moderator in the prediction of blood pressure obtained during five stressful tasks. *Journal of Research in Personality*, 31(4):523–542.

Liu, Y., Shriberg, E., Stolcke, A., Hillard, D., Ostendorf, M., and Harper, M. (2006). Enriching speech recognition with automatic detection of sentence boundaries and disfluencies. *Audio, Speech, and Language Processing, IEEE Transactions on*, 14(5):1526–1540.

Martineau, W. H. (1972). A model of the social functions of humor. *The psychology of humor: Theoretical perspectives and empirical issues*, pages 101–125.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781.

Miller, G. A. (1995). Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Okazaki, N. (2007). Crfsuite: a fast implementation of conditional random fields (crfs). *URL http://www. chokkan. org/software/crfsuite*.

Pascanu, R., Mikolov, T., and Bengio, Y. (2012). On the difficulty of training recurrent neural networks. *arXiv preprint arXiv:1211.5063*.

Pascanu, R., Gulcehre, C., Cho, K., and Bengio, Y. (2013). How to construct deep recurrent neural networks. *arXiv preprint arXiv:1312.6026*.

Rakov, R. and Rosenberg, A. (2013). " sure, i did the right thing": a system for sarcasm detection in speech. In *INTERSPEECH*, pages 842–846.

Reyes, A. and Rosso, P. (2012). Making objective decisions from subjective data: Detecting irony in customer reviews. *Decision Support Systems*, 53(4):754–760.

Reyes, A., Rosso, P., and Veale, T. (2013). A multidimensional approach for detecting irony in twitter. *Language Resources and Evaluation*, 47(1):239–268.

Riloff, E., Qadir, A., Surve, P., De Silva, L., Gilbert, N., and Huang, R. (2013). Sarcasm as contrast between a positive sentiment and negative situation. In *EMNLP*, pages 704–714.

Schuller, B., Steidl, S., Batliner, A., Burkhardt, F., Devillers, L., Müller, C. A., and Narayanan, S. S. (2010). The interspeech 2010 paralinguistic challenge. In *INTERSPEECH*, pages 2794–2797.

Sumners, A. D. (1988). Humor: coping in recovery from addiction. *Issues in mental health nursing*, 9(2):169–179.

Wang, M. and Manning, C. D. (2013). Effect of nonlinear deep architecture in sequence labeling. In *Proceedings of the 6th International Joint Conference on Natural Language Processing (IJCNLP)*.

Yang, D., Lavie, A., Dyer, C., and Hovy, E. (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.

Zhang, J. J. and Fung, P. (2012). Automatic parliamentary meeting minute generation using rhetorical structure modeling. *Audio, Speech, and Language Processing, IEEE Transactions on*, 20(9):2492–2504.