HumorHawk at SemEval-2017 Task 6: Mixing Meaning and Sound for Humor Recognition

David Donahue, Alexey Romanov, Anna Rumshisky
Dept. of Computer Science
University of Massachusetts Lowell
198 Riverside St, Lowell, MA 01854
david_donahue@student.uml.edu
{aromanov, arum}@cs.uml.edu

Abstract

This paper describes the winning system for SemEval-2017 Task 6: #HashtagWars: Learning a Sense of Humor. Humor detection has up until now been predominantly addressed using feature-based approaches. Our system utilizes recurrent deep learning methods with dense embeddings to predict humorous tweets from the @midnight show #HashtagWars. In order to include both meaning and sound in the analysis, GloVe embeddings are combined with a novel phonetic representation to serve as input to an LSTM component. The output is combined with a character-based CNN model, and an XGBoost component in an ensemble model which achieved 0.675 accuracy in the official task evaluation.

1 Introduction

Computational approaches to how humour is expressed in language have received relatively limited attention up until very recently. With few exceptions, they have used feature-based machine learning techniques (Zhang and Liu, 2014; Radev et al., 2015) drawing on hand-engineered features such as sentence length, the number of nouns, number of adjectives, and tf-idf-based LexRank (Erkan and Radev, 2004). Among the recent proposals, puns have been emphasized as a crucial component of humor expression (Jaech et al., 2016). Others have proposed that text is perceived as humorous when it deviates in some way from what is expected (Radev et al., 2015). One of the reasons for such dominant position of the feature-based approaches is the fact that the datasets have been relatively small, rendering deep learning methods ineffective. Furthermore, existing humour detection datasets tended to treat humor as a classification task in which text has to be labeled as funny or not funny, with nothing in between, which makes the task considerably simpler. In contrast, the #HashtagWars dataset (Potash et al., 2016b) provided for SemEval-2016 Task 6 assumes that humor can be evaluated on a scale, reflecting the reality that humor is non-binary and some things may be seen as funnier than others. It is also large in size, making it better suited to the application of deep learning techniques.

SemEval 2017 Task 6 used the tweets posted by the viewers of the Comedy Central's @midnight show, the #HashtagWars segment. Our team participated in subtask A, which was as follows: given a pair of tweets supplied for a given hashtag by the viewers, the goal was to identify the tweet that the show judged to be funnier (Potash et al., 2017). This paper describes the winning submission, and specifically, our systems that took first and second place in the official rankings for the task.

Our goal was to create a model that could represent both meaning and sound, thus covering different aspects of the tweet that might make it funny. Word embeddings have been used in a variety of applications, but phonetic information can provide new insights into the punchline of humor not present in traditional embeddings. The pronunciation of a sentence is important to the delivery of a punchline, and can connect sound-alike words.

In our first submission for Subtask A, semantic information for each word is provided to the model in the form of a GloVe embedding. We then provide the model with a novel phonetic representation of each word, in the form of a learned phonetic embedding taken as an intermediate state from an encoder-decoder character-to-phoneme model. With access to both meaning and

Proceedings of the 11th International Workshop on Semantic Evaluations (SemEval-2017), pages 98–102,
Vancouver, Canada, August 3 - 4, 2017. ©2017 Association for Computational Linguistics

sound embeddings, the model learns to read each tweet using a Long Short-Term Memory (LSTM) Recurrent Neural Network (RNN) encoder. The encoded state of each tweet passes into dense layers, where a prediction is made as to which tweet is funnier.

In addition to the embedding model described above, we construct a Convolutional Neural Network (CNN) to process each tweet character by character. This character-level model was used by Potash et al. (2016b), and serves as a baseline. The output of the CNN feeds into the same final dense layers as the embedding LSTM tweet encoders. This model achieved 63.7% accuracy in the official task evaluation, placing it second in the official task rankings.

To boost prediction performance further, we built an ensemble model over different model configurations. In addition to the model above, we provided an embedding-LSTM-only model and a character-CNN-only model as input to the ensemble. Inspired by previous work in NLP, we added an XGBoost feature-based model as input to the ensemble. This system was our second submission. The predictions of the ensemble model achieved 67.5% accuracy, placing it first in the official rankings for the task.

We also report experiments we conducted after the release of the test data, in which a few of the bugs present in the original submissions were addressed, and in which the best model achieves the accuracy of 68.3%.

2 Previous Work

Considerable research has gone into understanding the properties of humor in text. Radev et al. (2015) used a feature-bucket approach to analyze captions from the New Yorker Caption Contest. They noted that negative sentiment, human-centeredness and lexical centrality were their most important model features. Zhang and Liu (2014) trained a classifier using tweets that use the hashtag #Humor for positive examples. They concluded that tweet part-of-speech ratios are a major factor in humor detection. They also showed that sexuality and politics are popular topics in Twitter jokes that can boost humor perception. Jaech et al. (2016) and Miller and Turković (2016) explored the complicated nature of puns and their role in humor. Barbieri and Saggion (2014) explored the concept of irony in humor and used a large variety of syntactic and semantic features to detect irony in tweets. To summarize, negative sentiment, human-centeredness, lexical centrality, syntax, puns, and irony represent just a few of many aspects that characterize humor in text.

The majority of attempts at humor detection, including those listed above, rely on hand-engineered features to distinguish humor from non-humor. However, recently deep learning strategies have also been employed. Chen and Lee (2017) used convolutional networks to make predictions on humorous/non-humorous sentences in a TED talk corpus. Bertero and Fung (2016) predicted punchlines using textual and audio features from the popular sitcom The Big Bang Theory. While feature-based solutions use linguistic properties of text to detect humour, our hope in experimenting with deep learning models for this task was that they could capture such properties in a more unstructured form, without pre-determined hand-engineered indicators.

3 System Description

In order to identify the funnier tweet in each pair, as required by the task setup, we build the following models:

- Character-to-Phoneme Model (C2P)
- Embedding Humor Model (EHM)
- Character Humor Model (CHM)
- Embedding/Character Joint Model (ECJM)
- XGBoost Feature-Based Model (XGBM)
- Ensemble Model (ENSEMBLE)

3.1 Character-to-Phoneme Model

In addition to understanding the meaning of each word in the sentence and how those meanings fit together, some words sound funnier to the ear than others. The sound of a sentence might also reveal the power of its punchline.

To give the model a representation of sound (i.e., pronunciation) for each word, we train an encoder-decoder LSTM model to convert a sequence of characters (via learned character embeddings) into a sequence of phonemes. Much like other sequence-to-sequence models, our model learns how to convert an English word into a sequence of phonemes that determine how that word is pronounced (see Figure 1).

We train and evaluate this model on the CMU Pronouncing Dictionary corpus (Lenzo, 2017), which contains mappings from each word to its

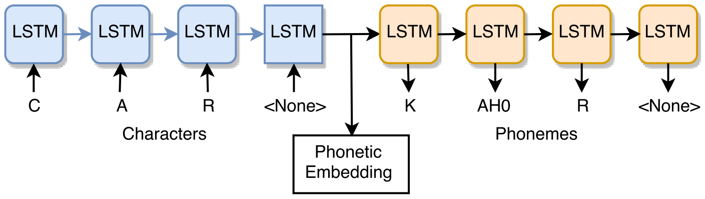

Figure 1: Character-to-Phoneme Model

corresponding phonemes. We use a 0.6/0.4 train-test split. Once the model is trained, we extract the intermediate embedding state (200 dim) between the encoder and decoder; this acts as a phonetic embedding, containing all information needed to pronounce the word. The resulting phonetic embedding for each word is concatenated with a semantic embedding to serve as the input for the embedding humor model (see below). Table 3.1 shows sample output of the model.

3.2 Embedding Humor Model

For both tweets in a tweet pair, a concatenation of a GloVe word embedding (Pennington et al., 2014) and phonetic embedding is processed by an LSTM encoder at each time-step (per word). We use word embeddings pre-trained on a Twitter corpus, available on the GloVe website[1]. Zero padding is added to the end of each tweet for a maximum length of 20 words/tweet. The output of each LSTM encoder (800 dim) is inserted into dense layers, and a binary classification decision is generated.

3.3 Character Humor Model

The character-based humor model processes each tweet as a sequence of characters with a CNN (Koushik, 2016). 30-dimensional embeddings are learned per character as input. The output of the CNN for both tweets in the pair are inserted into dense layers.

3.4 XGBoost Feature-Based Model

In order to approach the problem from a different prospective, in addition to the neural network-based systems described above, we constructed a feature-based model using XGBoost (Chen and Guestrin, 2016). In line with previous work (Radev et al., 2015; Zhang and Liu, 2014), we used the following features as input to the model:

1. Sentiment of each tweet in a pair, obtained with TwitterHawk, a state-of-the-art sentiment analysis system for Twitter (Boag et al., 2015).
2. Sentiment of the tokenized hashtag.
3. Length of each tweet in both tokens and characters (a very long tweet might not be funny)
4. Distance of the average GloVe embeddings of the tokens of the tweets to the global centroid of the embeddings of all tweets for the given hashtag.
5. Minimum, maximum and average distance from each token in a tweet to the hashtag.
6. Number of tokens belonging to the top-10 most frequent POS tags on the training data.

3.5 Embedding/Character Joint Model

The output of the embedding model LSTM encoders and the character model CNN encoders are fed into dense layers. For encoder input N, the three dense layers are of size $(3/4)N$, $(1/2)N$, and 1. Each layer gradually reduces dimensionality to final binary decision.

3.6 Ensemble Model

Inspired by the success of ensemble models in other tasks (Potash et al., 2016a; Rychalska et al., 2016) we built an ensemble model that combines the predictions of the character-based model, embedding-based model, the character/embedding joint humor model, and the feature-based XGBoost model to make the final prediction which incorporates different views of the input data. For the ensemble model itself, we use an XGBoost model again. Input predictions are obtained by using 5-fold cross-validation on the training data.

4 Results

Accuracies are calculated over three run average. Embedding/character models trained for five epochs with a learning rate of 1e-5 using the Adam optimizer (Kingma and Ba, 2014). Parameters are

[1] https://nlp.stanford.edu/projects/glove/

Word	Model Output	CMU Dictionary
rupard	R UW0 P ER0 D D	R UW1 P ER0 D
disabling	D AY1 S EY1 B L IH0 NG	D IH0 S EY1 B AH0 L IH0 NG
clipping	K L IH1 P IH0 NG	K L IH1 P IH0 NG
enfranchised	IH0 N F R AE1 N SH AY2 D D	EH0 N F R AE1 N CH AY2 Z D
eimer	AY1 M ER0	AY1 M ER0
dowel	D AW1 AH0 L	D AW1 AH0 L
vasilly	V AE1 S IH0 L IY0	V AH0 S IH1 L IY0

Table 1: Sample character-to-phoneme model output.

Model Configuration/Features	Trial Acc	Evaluation Acc	Official Evaluation Acc
ENSEMBLE	64.02%	65.99 %	**67.5%** (Run #2)
ECJM	59.31%	**68.30%**	63.7% (Run #1)
ECJM (GloVe-only)	64.42%	65.95%	
EHM	58.09%	67.56%	
EHM (GloVe-only)	**64.76%**	67.44%	
EHM (Phonetic-only)	54.55%	65.93%	
CHM	59.59%	63.52%	
XGBM	57.02%	60.35%	

Table 2: Model performance (accuracy). Official results reported for joint and ensemble models.

tuned to the trial set, which contained five hashtags. Train, trial and evaluation datasets were provided by task organizers, with the evaluation data containing six hashtags. Table 2 shows the results obtained by different models on the evaluation data. Note that the reported figures were obtained in additional experiments after a few of the bugs present in the original submission were addressed. For completeness, we also report the official results obtained by our system submissions (runs #1 and #2).

5 Discussion

The ensemble model performed the best during the official evaluation, placing it 1st among 10 runs, submitted by the 7 participating teams. Note that accuracies on evaluation hashtags are on average 5.36% higher than on trial hashtags (see Table 2). This suggests each dataset contains different hashtag types, and that the evaluation set more closely matches the training set. For example, phonetic embeddings reduce performance in the trial set and improve performance in the evaluation set. We hypothesize that phonetic embeddings are not important for some hashtags, and that the evaluation set contains more such hashtags .

While adding phonetic embeddings and/or the character model yields inconsistent results across the trial and evaluation sets, adding the GloVe representation produced the best scores for both datasets. From these results, token-based semantic knowledge appears to be the most important factor in humor recognition for this dataset. These results differ from that of Potash et al. (2016b), who report that a CNN-based character model achieves the highest accuracy on leave-one-out evaluation.

The character-to-phoneme model yields very interesting results upon testing. The model correctly classifies 75% of phonemes in the test set. As shown in Table 3.1, the model often guesses a similar-sounding phoneme in cases when the correct phoneme is not guessed. For example, in ’vasilly’, AE1 is guessed instead of AH0.

6 Conclusion

The learned character embeddings achieved reasonable results on both trial and evaluation data. The incorporation of phonetic embeddings in humor prediction, on the other hand, appears to yield inconsistent performance across different hashtags. The ensemble model improved performance on the official data. Overall, GloVe embeddings consistently improved performance, highlighting the importance of lexical semantic information for this humour classification task.

References

Francesco Barbieri and Horacio Saggion. 2014. Automatic detection of irony and humour in twitter. In *Proceedings of the International Conference on Computational Creativity*.

Dario Bertero and Pascale Fung. 2016. Deep learning of audio and language features for humor prediction. In *International Conference on Language Resources and Evaluation (LREC)*.

William Boag, Peter Potash, and Anna Rumshisky. 2015. Twitterhawk: A feature bucket approach to sentiment analysis. *SemEval-2015* page 640.

Lei Chen and Chong MIn Lee. 2017. Convolutional neural network for humor recognition. *arXiv preprint arXiv:1702.02584* .

Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, pages 785–794.

Günes Erkan and Dragomir R Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research* 22:457–479.

Aaron Jaech, Rik Koncel-Kedziorski, and Mari Ostendorf. 2016. Phonological pun-derstanding. In *Proceedings of NAACL-HLT*. pages 654–663.

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

Jayanth Koushik. 2016. Understanding convolutional neural networks. *arXiv preprint arXiv:1605.09081* .

Kevin Lenzo. 2017. The cmu pronouncing dictionary. https://doi.org/http://www.speech.cs.cmu.edu/cgi-bin/cmudict.

Tristan Miller and Mladen Turković. 2016. Towards the automatic detection and identification of english puns. *The European Journal of Humour Research* 4(1):59–75.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*. volume 14, pages 1532–1543.

Peter Potash, William Boag, Alexey Romanov, Vasili Ramanishka, and Anna Rumshisky. 2016a. Simihawk at semeval-2016 task 1: A deep ensemble system for semantic textual similarity. *Proceedings of SemEval* pages 741–748.

Peter Potash, Alexey Romanov, and Anna Rumshisky. 2016b. # 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)*. Association for Computational Linguistics.

Dragomir Radev, Amanda Stent, Joel Tetreault, Aasish Pappu, Aikaterini Iliakopoulou, Agustin Chanfreau, Paloma de Juan, Jordi Vallmitjana, Alejandro Jaimes, Rahul Jha, et al. 2015. Humor in collective discourse: Unsupervised funniness detection in the new yorker cartoon caption contest. *arXiv preprint arXiv:1506.08126* .

Barbara Rychalska, Katarzyna Pakulska, Krystyna Chodorowska, Wojciech Walczak, and Piotr Andruszkiewicz. 2016. Samsung poland nlp team at semeval-2016 task 1: Necessity for diversity; combining recursive autoencoders, wordnet and ensemble methods to measure semantic similarity. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*. Association for Computational Linguistics, San Diego, California, pages 602–608. http://www.aclweb.org/anthology/S16-1091.

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, pages 889–898.