Humor & Sarcasm Detection

William Jiang
New York University
New York, NY 10003
wj419@nyu.edu

Cody Gilbert
New York University
New York, NY 10003
cjg507@nyu.edu

Abstract

This project sought to generalize and improve upon existing humor recognition methods on a corpus of sarcasm data using deep learning classification models. This project evaluated various state-of-the-art models including BERT and custom multi-layer recurrent neural networks with features created from text data. Of the adhoc features considered, an estimated surrogate of textual semantic conflict built from language model loss was proposed. The results of this project show little success in generalizing humor recognition using deep learning methods, and that the proposed semantic conflict estimate variable gives no statistically significant value in end-to-end humor classification.

1 Introduction

In 2011, a memorial dedicated to the late US civil rights leader Martin Luther King Jr. was unveiled to the public beside the Potomac River in the nation's capital (Miller [8]). This monument was crafted from monolithic white marble boulders, into which selected quotes from MLK Jr.'s written works on human rights were carved. One of the engraved statements sought to paraphrase the following:

> "Yes, if you want to say that I was a drum major, say that I was a drum major for justice. Say that I was a drum major for peace. I was a drum major for righteousness. And all of the other shallow things will not matter."

However, in the misfortunate attempt to shorten this quote, the sculpture carved the statement

> "I was a drum major for justice, peace and righteousness"

The paraphrased quote appears to assert the opposite meaning intended by MLK Jr.'s original quote: that he rejected the title of "drum major" as a caricature of his efforts, but that if the title were used then it should include the greater purpose of his work.

Significant resources were spent to correct the sculptor's error due to the mistranslation of a hypothetical assertion. This experience shows that the mistranslations of statements intended to convey absurdity or sarcastic comparison, as opposed to their literal translations, can create real-world complications. Furthermore, when natural language is modeled to derive the semantic meaning of the text, these humorous, sarcastic, and absurd statements must be identified to prevent extracting contradictory conclusions.

Natural human communication makes heavy use of this kind of humor, satire, and sarcasm to make complex statements and assertions, and the intent of this project is to create a natural language model that can classify these statements. This project will apply current and popular machine learning tools to create a method of dividing statements into the categories of "humorous" or "serious" as a binary classification task.

2 Previous Work

Humor classification or the automatic generation of humorous text has had a surprisingly rich background of research. An early paper on automatic humor recognition by Mihalcea and Strapparava [7] sought to classify simple "one-liners" taken from web-scrapings. Their methods considered only the text of a given joke without an analysis of tone or proceeding narrative, using both structural features (i.e. Naive Bayes and SVMs) and heuristic features that identified the humorous passages by alliteration, antonymy, and adult slang. Their results using a combination of the features gave a maximum 96.95% accuracy on a set of humorous text and an 84.83% accuracy on non-humorous data sets.

However, this work is limited in its applicability, as future work by Yang et al. [13] note that classifiers built with negative samples outside of the domain of positive examples can produce high accuracy with limited generality. In other words, the structure of a one-liner joke and a news title differ in more ways than simply being humorous. In Yang et al. [13], a general strategy of overcoming the differences in domain structure between positive and negative classes was to select only positive-negative sample pairs that contained the same words, and restrict negative samples to those of the same length as the positives.

More recent work by Bertero and Fung [5] sought to classify "punchlines" in the text of a sitcom using textual and acoustic features. These methods differ from pure humor-vs-non-humor classification in that they attempt to use the greater context of the script to determine a statement defining the locus of humor for a given setup. Their experiments show that using convolutional neural networks, recurrent neural networks, and conditional random fields on a joke's lexical features can provide modest improvement in accuracy over the baseline.

Yang et al. [13] improve on previous work by defining four humor features based on latent sentence structures: ambiguity, interpersonal effect, incongruity, and phonetic style. The paper's results show that these features created with a Word2Vec embedding vector and modeled with KNN give an accuracy of 85.4%. The authors also re-evaluated the Mihalcea and Strapparava [7] model adjusting for domain inconsistencies and achieved a 76% accuracy.

The latest work by Bali et al. [4] clusters humor into categories and subcategories based on computationally detectable characteristics in which the categories of humor recognized by the previous studies reside. Bali et al. [4] first construct models to classify text as a category of humor, and use a humor classification model built for the category using category-specific features. The results of their modeling show an accuracy range of 64-88%, depending on the humor category.

3 Methodology

3.1 Classification Approach

The previous works demonstrate a high degree of variance in accuracy based on the domain from which the humorous text is drawn and the categorization of humor on which hand-crafted features are created. These previous studies may note a general theories of humor, but like Bali et al. [4] divide humor into computationally classifiable categories to raise modeling accuracy at the expense of generalizability.

This project will use larger, more computationally intensive models such as BERT (Devlin et al. [6]) and deep recurrent neural networks to generalize these approaches. These large scale models are expected to implicitly learn the categorizations of humor that were explicitly approximated in prior work using hand-crafted features and clustering a-priori.

3.2 General Humor Features

While the main thrust of this project is to use deep learning to approximate the structures of humorous text in classification, further features of the text will be created based on humor theory to supplement model performance. The generalization this project takes to humor classification is built upon the insights gleaned by Bali et al. [4] via the Script Semantic Theory of Humour (SSTH) introduced by Victor Raskin (Raskin [10]) in which a verbal joke is defined by two conditions:

1. "The text is compatible, fully or in part, with two different (semantic) scripts"
2. "The two scripts with which the text is compatible are opposite. The two scripts with which the text is compatible are said to overlap fully or in part on this text."

In other words, a joke is defined as a sequence of text in which two opposing semantic meanings are posited. For example, we can look at a famous joke by Groucho Marx,

> "One morning I shot an elephant in my pajamas. How he got into my pajamas I'll never know."

In this example, the semantic meanings are that the man in question is both shooting an elephant while the man is wearing his own pajamas, that the elephant itself is wearing his pajamas, and that the elephant had the capacity to wear his pajamas in the first place. These semantic meanings are all incongruent, thus by the SSTH definition this text is humorous.

A feature that quantifies the amount by which multiple semantic meanings differ within a given portion of text is difficult to express verbally, let alone derive quantitatively. However, there exist other language abstraction tools that can be used to approximate the amount deviation a sentence has from a learned context; expressing the likelihood of a given sequence of text based on the context of a greater repository of text is the definition of a language model. If a language model can sufficiently capture the semantics of language to generate the probability of a sequence of text, then an approximation of the deviation of the semantic meaning of a given sequence to that of the corpus on which the model is trained can inferred from the sequence probability.

In other words, a low sequence probability suggests a large deviation in the sequence's underlying semantics from that of the semantics captured from the model's training corpus in similar contexts, and that deviation may be indicative of humor.

The validity of this assumption will be evaluated by creating a "Non-sequitur" (NS) feature from each text defined as the language model prediction loss of each text sequence. The language model used in this project must be sufficiently sophisticated to learn a language's underlying semantics, therefore this project will use the large-scale pre-trained BERT language model (Devlin et al. [6]) to create loss features. This feature will be added to a subset of the models evaluated in this project to determine if humor classification can be improved with this hypothesis.

3.3 Additional Features

Additional features were considered to enhance model accuracy. While the core of this project is to create a model with as few hand-crafted features as possible, the use of obscene language was noted as a common and predictively significant feature of humor in previous studies (Mihalcea and Strapparava [7], Bali et al. [4]). The modeling effort cost of adding a scalar input to existing models that indicated the presence of obscene language is relatively trivial in comparison to potential accuracy gains. Therefore, it was considered an acceptable feature to evaluate.

The obscene language identifier feature (OI) is an integer value 1 if the text of the comment contains an obscene or vulgar word, and 0 otherwise. A word is considered vulgar if it is present in a common listing of vulgarities. In this project, the Offensive/Profane Word List provided by Luis von Ahn's research group at Carnegie Mellon University (von Ahn [12]) was used as the corpus of obscene language.

3.4 Classification Models

The following models were used in this project for humor categorization.

3.4.1 Naive-Bayes (NB)

NB with n-gram features is a popular tool used in almost all the previous studies on humor classification, and will provide baseline results. NB will be implemented with the popular NLTK package (NLT [2]) using 1-3-grams.

3.4.2 Recurrent Neural Networks (RNNs)

RNNs using vanilla RNN nodes and LSTM nodes will be used for simple encoding models and as a subset of more complex hybrid models. These models will be implemented with the Keras and Tensorflow open-source models (Ker [1]).

3.4.3 Convolutional Neural Networks (CNNs)

Similar to RNNs, 1-D CNNs with associated pooling and dense layers will be used as standalone models and as subsets of hybrid models. Like RNNs, these models will be implemented with the Keras and Tensorflow open-source models (Ker [1]).

3.4.4 BERT

A pre-trained BERT model (Trivedi [11]) with a classification head will be tuned and used directly for classification. BERT with a masked language model head (ber [3]) will be used to construction language model losses used as "Non-sequitur" features as described in the previous section.

3.4.5 Hybrid Feature Models

These models combine the adhoc features and combinations of the above neural network models in an iterative manner to find a solution with the highest prediction accuracy. Due to the iterative nature of these models, the final resulting model will be described in the results section.

3.5 Data Sources

The data sets used in this project will be from a collection of labeled corpora containing statements flagged as humorous or not. To supply a large enough corpus of training data to permit tuning deep-learning models, an assumption was made to equate sarcasm and humor. This assumption was considered appropriate as Bali et al. [4] denotes sarcasm as a subcategory of humor, and the majority of sarcastic comments indicate a strong relation between sarcasm and humor.

This project used the Sarcasm on Reddit (Ofer [9]) dataset comprising of about 1.4 million user comments self-tagged with a "/s" identifier to indicate that the comment is intended to be sarcastic. The set is labeled with a balanced set of sarcastic and non-sarcastic user posts.

3.6 Data Preparation

Due to memory limitations on the computational servers used for model fitting, the sarcasm data was limited to only 10,000 samples from the full 1.4 million sample dataset. Of these samples, a 25/75% test-training set split was created, and of the remaining training samples a 25/75% validation-training set split was created for model tuning. Biasing the training set to the first 10,000 rows skews the initially balanced training set to an approximately 63/37 negative to positive label ratio. To correct accuracy predictions, a vacuous model predicting all positive labels will be used to set the baseline accuracy to which subsequent models will be compared.

All text comments were cleaned to remove non-terminating punctuation (".", "?", and "!"). The input text to the NB model had additional cleaning to remove stop words and lemmatize individual words. Neural net models omitted word removal and stemming to prevent removing contextual elements such as tenses from the word sequences. Neural net models were given a vocabulary size limit of 5000 words and a maximum sequence length of 200 to fix the size of comment arrays being processed by the model.

4 Results

4.1 Hybrid Model

The most general hybrid model that converged over the design iterations is shown in Figure 4.1. The most general hybrid model consisted of a deep neural network including three features: comment word embeddings, OI, and NS. The word embeddings were processed through two bidirectional

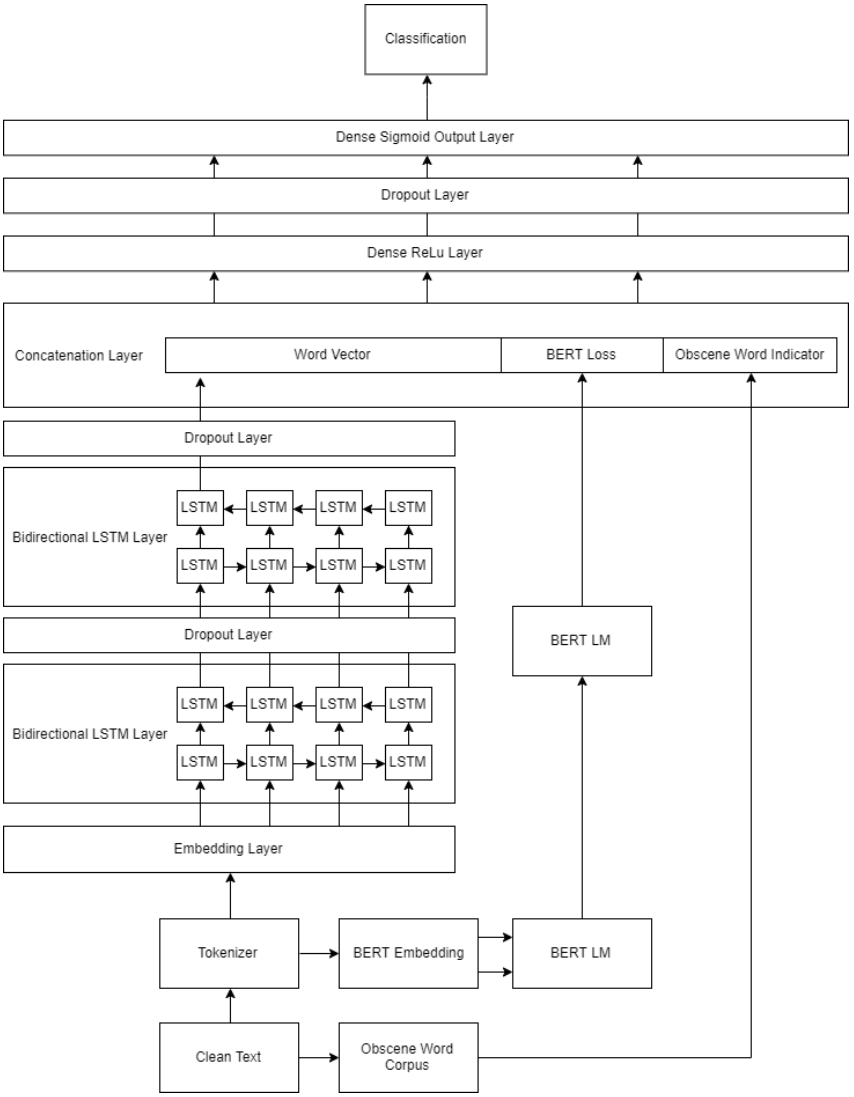

Figure 1: Hybrid Model with NS and OI Features

LSTM layers, with the first bidirectional LSTM layer feeding all intermediate states to the second, and the second emitting only the final encoding. The final encoded output vector is then concatenated to the NS and OI variable calculated as described in the previous section.

The concatenated feature vector was fed into a dense neural layer with ReLu activations to promote "mixing" of the input features. The output of this dense layer was passed to another dense sigmoid-activated layer used to predict the final binary classification. Each of the neural layers included a dropout layer with a 20% variable drop rate to prevent overfitting. These dropout layers were considered crucial as the models showed perfect training-set prediction at around 10 epoches without the dropout layers. Additional hybrid models were trained on combinations of the three features to assess their individual impact on prediction accuracy.

4.2 Overall Validation Accuracy

Table 1 shows the maximum validation set accuracy for all assessed models.

The results show that the models provide little improvement over the vacuous baseline, and well below the accuracy ranges established by previous work. The highest accuracy model was the Hybrid

Table 1: Overall Model Validation Set Accuracy

Model	**Accuracy**
Vacuous Model (All Positive)	63.2%
Naive Bayes (1, 2, 3-grams)	63.0%
BERT Classifier	63.0%
Hybrid LSTM - Text Only	67.4%
Hybrid LSTM - OI	**68.6%**
Hybrid LSTM - NS	68.3%
Hybrid LSTM - NS + OI	67.9%

Table 2: Hybrid LSTM - OI Model Values

Training Epochs	2
Embedding Vector Length	300
Dense ReLu Units	20
LSTM Cell Units	300
Test Set Accuracy	66.4%
Test Set Precision	54.5%
Test Set Recall	42.3%
Test Set AUC	68.3%

LSTM with word embeddings and OI features with a validation set accuracy of 68.6%. Table 2 shows the model parameter values and test set metrics of the Hybrid LSTM - OI model.

The validation set accuracy differences between the hybrid models with different feature subsets show a small degree of variance that may be explained by random noise in parameter initialization. However, an important conclusion from the feature combination results is that the NS feature drawn from language model loss provides no significant gains in model performance compared to the word embeddings alone. This outcome may either indicate that this additional information is contained in the word embeddings themselves, or that the hypothesised correlation of language model loss to semantic conflict is flawed in either theory or in its application here. In either case, this feature provides no significant predictive value.

The results found in this project show that prediction of humor as abstracted by the label of sarcasm is difficult to classify using deep neural networks. Improved results may be obtained by using more sophisticated models not considered here, using more training data, or using additional text features. In the absence of additional modeling iterations, it can be concluded that general humor and sarcasm classification is not easily generalized by deep learning methods.

5 Conclusion

This project sought to generalize existing humor recognition methods on a corpus of sarcasm data via the use of deep learning methods with limited adhoc features created from text data. Of the adhoc features considered, an estimated surrogate of textual semantic conflict built from language model loss was proposed. The results of this project show little success in generalizing humor recognition using deep learning methods, and that the proposed textual semantic conflict estimate variable gives no statistically significant value in end-to-end classification.

Acknowledgments

The team would like to thank the New York University Courant Institute of Mathematical Sciences (CIMS) compute server administration and support team for continued efforts in maintaining the server clusters used to make projects like this one possible.

References

[1] Keras: The python deep learning library. URL `https://keras.io/`.

[2] Natural language toolkit. URL `https://www.nltk.org/index.html`.

[3] Transformers. URL https://github.com/huggingface/transformers.

[4] Taradheesh Bali, Vikram Ahuja, and Navjyoti Singh. What makes us laugh? investigations into automatic humor classificationn. pages 1–9, 2018. URL https://www.aclweb.org/anthology/W18-1101.pdf.

[5] Dario Bertero and Pascale Fung. Deep learning of audio and language features for humor prediction. 2016. URL http://www.lrec-conf.org/proceedings/lrec2016/pdf/927_Paper.pdf.

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. 2018. URL https://arxiv.org/abs/1810.04805.

[7] Rada Mihalcea and Carlo Strapparava. Making computers laugh: Investigations in automatic humor recognition. page 531–538, 2005. URL https://pdfs.semanticscholar.org/89ff/2c1c6337345213886c1a5f0905008d08ea52.pdf#page=567.

[8] Jake Miller. Erroneous quote on mlk memorial to be removed. 2013. URL https://www.cbsnews.com/news/erroneous-quote-on-mlk-memorial-to-be-removed/.

[9] Dan Ofer. Sarcasm on reddit: 1.3 million labelled comments from reddit. 2018. URL https://www.kaggle.com/danofer/sarcasm.

[10] Victor Raskin. *Semantic Mechanisms of Humor*. 1984. URL https://link.springer.com/book/10.1007/978-94-009-6472-3.

[11] Kaushal Trivedi. Fast-bert. URL https://github.com/kaushaltrivedi/fast-bert.

[12] Luis von Ahn. Offensive/profane word list. URL https://www.cs.cmu.edu/~biglou/resources/.

[13] Diyi Yang, Alon Lavie, Chris Dyer, and Eduard Hovy. Humor recognition and humor anchor extraction. page 2367–2376, 2015. URL https://www.cs.cmu.edu/~alavie/papers/D15-1284.pdf.