SemEval-2020 Task 7: Assessing Humor in Edited News Headlines

Varsha Vatsavai, Humera Aamreen, Kathryn Young
Virginia Commonwealth University
{vatsavaishv, aamreenh, youngk6}@vcu.edu

Abstract

Computational humor analysis has seen much progress in classifying whether some piece of text is funny or non-funny. This paper presents a system for the regression task proposed in SemEval 2020: Assessing Humor in Edited News Headlines, to predict the mean funniness of a headline that is replaced with short edits to make it funny. We implement a recurrent neural network model trained from the averaged word embeddings of each headline to predict the funniness score. The BiLSTM model used memorizes the previous elements to capture the essential humor from a sentence. Results show that our model performs significantly better than the baseline model, despite the incongruity in humor structures and ratings of the dataset

1 Introduction

Understanding humor is a task that even humans struggle with sometimes. It requires an understanding of world-knowledge, common sense, and the ability to perceive relationships across entities and objects. There have been advances in humor recognition, but generation has had much less progress. The overarching goal of our work is to estimate the funniness of news headlines that have been modified using a micro-edit to make them funny. Micro-edits in our study mean one word is replaced by another (i.e. an entity replaced by a noun, noun replaced by a noun, and verb replaced by a verb). Edited headlines in the dataset are scored by 5 judges who rate the headline from 0 (not funny) to 3 (funny). The mean of a headline's grades is considered the ground truth funniness.

Traditional approaches of humor detection assume a single, unambiguous word in a document. However, humor involves the single word to be interpreted as multiple implications and different understanding of the same sentence. When a computer converses with human being and if it can detect the humor in human's language, it can be considered as a significant evolution in AI. This allows the computer to make better decisions and improve user experience. Thus, developing techniques to enable computers to understand humor in human conversations and adapt behavior accordingly deserves particular attention.

Humor influences memory and attention, it enables social interaction and enhances communication. For computers to effectively communicate with humans or serve as a practical model of human language understanding, they should also be competent in dealing with humor. Computational humor detection has therefore evolved as a challenging AI problem. The dataset for our research in computational humor, Humicroedit, is obtained from the original news headlines posted on Reddit, which are edited by qualified annotators from Amazon Mechanical Turk to generate

From the collection of news headlines, the elementary task following is to edit these headlines to create a micro-edit, which produces a smallest change in the headline while inducing the essential punch of humor. The micro-edits effective in generating humor in the headlines are created by the editors through multiple ways that suit the headline. Creating sarcasm, replacing a headline word with its strongly connected substitute, belittling an entity in the headline and adding punchlines are few of the strategies adopted to add humor to the micro-edits. Furthermore, to achieve the goal of modeling computational humor, baseline classifiers are to be developed to predict if the edited headline is funny and correspondingly determine its mean funniness. And also, given a general headline and two of its edited versions, a classifier that predicts the funnier version.

2 Related Work

Chen and Soo (2018) implemented a Convolutional Neural Network (CNN) to detect humor within four datasets [1]. They conducted their research using *Pun of the Day*, *One Liners*, *Short Jokes,* and *PTT Jokes*. The datasets had diverse looking data all with different types of humor, sentence lengths, data sizes, and languages. The experimenters thought CNN could be successful due to the fact that it has already had success in other text categorization tasks.

The study also looks at increasing depth using the concept of a *Highway Network*. The *Highway Network* allows shortcut connections with gate functions. The gate units regulate the matriculation of *information* through the network, allowing for the use of increasingly deeper nets. After going through the nets, dropout is used, and results come up through the output layer. Using this method, the model was able to outperform previous works in terms of accuracy, precision, recall, and F1 scores. All the works mentioned in this paper note that CNN models have done better than other methods, but with the additional features in this experiment, the results improve even further. The classifications of this experiment are binary (i.e. funny or not funny) unlike our SemEval Task which rates sentences on a funniness scale.

Bertero and Fung propose Long Short-Term memory-based framework to predict humor [2]. They analyzed data from a popular TV-Sitcom, whose canned laughter gives an indication of when the audience would react. They modeled the *setup*-punchline relation of conversational humor with a Long Short-Term Memory, with utterance encodings obtained from a CNN. Their framework performed well compared to the CRF baseline approach.

Humor recognition has seen some progress in recent times, while automatic humor generation is still a research problem. [3] conducted a survey from professional comedians and found evidence that humor-generation process can be described which helped in the progress towards the fully automated generation of humor. They analyzed news satire from The Onion and decomposed the process of humor creation into seven microtasks ranging from the identification of entities and aspects in an input headline to articulating associations and underlying beliefs. They developed a workflow, inspired by the design literature, that invokes these microtasks in a novel, dynamic manner following four design principles: Understanding the problem, Ideation, applying solution patterns, and evaluation. 85% of the evaluators found that the workflow made their process more methodical and the microtasks enabled them to make a wider variety of jokes.

One of the most recent works on humor analysis, similar to our work on humor ranking for edited funny headlines is by Altin et al. (2019), who performed a classification to assess whether tweets in Spanish are funny or not [4]. The analysis was carried out through a multi-task learning approach using bidirectional long short-term memory (biLSTM) models. The output from this classification task was fed to another output layer, where the funniness scores were assigned to a given tweet in a 5-star ranking. Another approach for the same task by Mao and Liu (2019), used BERT, a multi-layer bidirectional transformer encoder to learn bi-directional representations for training [5]. Research on humor ranking is also carried out in the shared task 'SemEval-2017 Task 6: HashtagWars: Learning a sense of humor' where tweets are ranked based on how funny they are [6]. The best ranked system for this task used an ensemble of feature-based and neural network-based system, which infers that human intuition, besides neural networks, is essential for automated understanding of humor.

Humor analysis in edited news headlines is also carried out by West and Horvitz (2019), with the difference in generating edited headlines [7]. They used a Web-based game to build a corpus of similar and satirical pairs, to analyze the words that cause a switch from serious to funny. Humorous satirical headlines are examined by Skalicky (2018) to assess whether the use of satire deviates from expected language patterns [10]. The results from this analysis suggest that many satirical headlines violate the linguistic expectations that in turn helps the person on the other side to interpret humor and satire easily. This study suggests that the headlines need to rely on these deviations from linguistic patterns so as to direct a reader towards a satirical message.

Research on automated humor contains a number of binary classification problems that distinguish humor from non-humor. Yang et al. (2015) developed a humor recognizer that effectively distinguishes humor and non-humor texts, while also extracting the anchors that validate humor in a sentence [8]. Zhang and Liu (2014) designed humor-related features and used machine learning techniques to distinguish humorous tweets from non-humorous ones [9].

3 Dataset

Humicroedit is the dataset we use for our experiments. The dataset contains real news headlines, in English, gathered from Reddit. Expert annotators were qualified from Amazon Mechanical Turk to make edits they perceived as humorous to the headlines, and each edited headline had its funniness judged by five annotators. The resulting dataset contains 15,095 headlines along with their edits and an average rating the edit's humor.

Humicroedit presents new opportunities in the field of computational humor for many reasons. The headlines in the dataset are not based on a template, just as headlines in the real world. To determine the funniness of headlines in this dataset, an understanding of world-knowledge and common sense is necessary. The humor proposed by the editors is sometimes only understood through multiple layers of cognition, meaning that creating a model to predict humor from these headlines is not a simplistic and arbitrary task. This dataset presents the need for development of more advanced NLP tools that can recognize patterns but also use semantic understanding and reasoning. Some judges may think one thing is funny, other judges might not think it is. The headlines are annotated by different combinations of judges, which can skew the scores. For example, when there are two similar headlines, the set of judges for the first headline might think the headline is funny, while the set of judges for the second headline might not be amused.

4 Methods

A. Preprocessing

To preprocess our data, we first removed entries in the dataset with zero grades, since these entries produced a meanGrade of 0, while in fact that meanGrade isn't the actual assessment of the humor in the headline. Keeping these entries in our evaluation would throw off our analysis. After this, features that were deemed unnecessary were dropped from the data set, such as "id" and "grades". These values weren't destined to be part of the analysis.

Next, we inserted the edit word into its proper place in the headline and stored the edited headline as a new column in the dataset called "replaced_sentence". Every letter in the replaced sentence was converted to be lowercase, for consistency sake. Stop words and punctuation were also removed from "replaced_sentence". We then removed the strings 's' and 'nt' from sentences since they were perceived to be their own words, when really, they were part of a contraction or a possession and got separated once punctuation was removed. Then, we also removed the ten most rare words from sentences as well as digits. Spell check was applied as well, even though the likelihood of misspellings was low since the data came from actual news headlines. We also applied lemmatization on our data as a form of normalization**.** Once this preprocessing was all done, the processed sentences were tokenized and stored as a new column in the data set.

B. Feature Extraction

After the preprocessing step, we convert the words in each sentence into tokens using nltk library. Next step is to convert the tokens to word embeddings. We used pre-trained word embeddings to convert the tokens to word embeddings. Pre-trained models are the simplest way to start working with word embeddings. A pre-trained model is a set of word embeddings that have been created elsewhere that can simply load onto our computer and memory. For our task we used pre-trained word embedding model in Gensim which is trained on Google News Dataset as our task dataset is related to News Headlines.

After extracting the word embeddings of each word individually, sentence embeddings are created in the form of a one 300-dimension vector that represents the whole sentence. These sentence embeddings are used to train the neural

network model which is discussed in later in this section.

C. Models

For the tasks of computational humor, there are many approaches that exploit the learning capabilities of Neural Networks through CNN or RNN. We model the task of estimating the mean funniness grade of edited news headlines by using RNNs, which make use of sequential information to capture the meaning of words. CNNs do not contain any memory associated with the model for processing sequential information and RNNs are therefore highly suitable for processing the sequences that require memorizing of previous elements to capture the complete meaning. We deploy a special kind of RNN, LSTM network to model the edited news headlines in the funniness prediction.

Given a sequence of tokens from the edited news headlines, we get their embeddings as a sequence of vectors and feed them into the LSTM unit, which reads each input sequentially and updates the hidden state, holding the summary of the processed information. We then use a BiLSTM to obtain the summarizations from both directions. A BiLSTM contains a forward LSTM and a backward LSTM that reads the sentences in both directions and concatenates their annotations.

We use this BiLSTM model with a sigmoid activation function and Adam optimizer using the Keras library. We also add the dropout layers to prevent the overfitting of training data.

5 Evaluation

The evaluation metric for our task is Root Mean Square Error (RMSE), which measures the sample standard deviation of differences between the predicted values and the observed values. Smaller RMSE corresponds to better performance of the model. It measures the regression fit of the model and is therefore useful in evaluating our regression task of predicting the mean funniness grade of the edited news headlines.

6 Results

We now investigate the performance of the LSTM and BiLSTM models trained on the test dataset. LSTM model with the sigmoid activation function produced RMSE of 0.557, While BiLSTM model with sigmoid activation function achieved a significantly better RMSE 0.556 and with the Relu activation function the RMSE increased to 0.78 which is not desirable for the task. Hence the best system is the BiLSTM model with sigmoid activation layer.

We also tried a linear regression model, since our task is a regression task. We fed in the sentence vectors for each sentence (which were 300-dimensions each). The result of this method was an RMSE of 0.5571. This is just slightly better than the best biLSTM model (RMSE of 0.5575). It also took much less time to train. However, we decided that the best model was the bi-LSTM, since there is more room for continued tuning of the model, and it performed better than the baseline model whose RMSE is 0.5784. This is the reason we chose to carry out our validation on the biLSTM method.

Hyperparameter Tuning plays a very important role in the performance of the model. We have considered optimizing various parameters like number of layers, activation function, optimizer, batch size, epochs, dropout rate, learning rate etc. while tuning our dense network.

7 Conclusion

The existing computational humor analysis techniques essentially involve the study of whether a section of text is funny. This task analyzes humor in short edits made to a text that change from funny to non-funny. We used the pretrained word embeddings from Google News Headlines to obtain the word vector representations, which were averaged on each headline to be used as input to our recurrent neural network model. We use BiLSTM due its ability to capture the memory units from a sequence of word vectors that helps in better analyzing the humor traits. Although not the best ranked thus far, our model performed better than the baseline model with a better RMSE score. Given the complexity of the humor structures and their ratings, we are pleased with the ability of our system to capture the humor of the edited headlines.

References

1.Chen, Peng-Yu, and Von-Wun Soo. “Humor Recognition Using Deep Learning.” ACL Anthology, 2018, pp. 113–116.

2.Dario Bertero, and Pascale Fung. “A Long Short-Term Memory Framework for Predicting Humor in Dialogues.” NAACL-HLT, 2016, pp.130-135

3.Lydia B. Chilton, James A. Landay and Daniel S. Weld. “HumorTools: A microtask Workflow for Writing News Satires”

4.Altin, Lutfiye Seda Mut, Àlex Bravo, and Horacio Saggion. “LaSTUS/TALN at HAHA: Humor Analysis Based on Human Annotation,” 2019, 6.

5.Mao, Jihang, and Wanli Liu. “A BERT-Based Approach for Automatic Humor Detection and Scoring,” 2019, 6.

6.Potash, Peter, Alexey Romanov, and Anna Rumshisky. “SemEval-2017 Task 6: #HashtagWars: Learning a Sense of Humor.” In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, 49–57. Vancouver, Canada: Association for Computational Linguistics, 2017

7.West, Robert, and Eric Horvitz. “Reverse-Engineering Satire, or ‘Paper on Computational Humor Accepted despite Making Serious Advances,’” n.d., 8.

8.Yang, Diyi, Alon Lavie, Chris Dyer, and Eduard Hovy. “Humor Recognition and Humor Anchor Extraction.” In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2367–76. Lisbon, Portugal: Association for Computational Linguistics, 2015.

9.Zhang, Renxian, and Naishi Liu. “Recognizing Humor on Twitter,” n.d., 10.

10.Skalicky, Stephen. “Lexical Priming in Humorous Satirical Newspaper Headlines.” *HUMOR* 31, no. 4 (2018): 583–602

11.Hossain, Nabil, et al. “‘President Vows to Cut Hair’: Dataset and Analysis of Creative Text Editing for Humorous Headlines.” *Proceedings of the 2019 Conference of the North*, June 2019