SI630 Final Project Proposal
Humor Detection

Xinhao Liao

Abstract

Funniness detection and generation have always been a challenge for researchers. **Humicroedit**, a new dataset introduced in (Hossain et al., 2019), contains news headlines specifically edited and then manually evaluated based on funniness. *SemEval-2020 Task 7: Assessing Humor in Edited News Headlines* of *CodaLab*, posted in https://competitions.codalab.org/competitions/20970, raises the problem of regression of the average funniness grades. The tasks aim to study the humorous effects of edits and bring more in-depth understanding of humor detection and generation.

1 Introduction

Humor and funniness is everywhere in text and conversations. Yet this can be difficult for machines, and even human beings some time, to detect and create humor, since it generally requires in-depth understanding of the context as well as some creativity.

Compared to humor generation, humor detection is a much easier task and can bring more insights about human generation patterns and researches. And to learn more from humor detection, a binary classification problem might be not enough, and we might be able to learn more about the intensities of humor from a regression problem which gives some grades about the funniness.

Previous tasks and researches like (Castro et al., 2018) and (Potash et al., 2017) tend to study such problems from raw text data like tweets. A new dataset introduced by (Hossain et al., 2019) and later described in section 3 brings focus to the problem of humorous effects of short edits and replacements in the text.

Beyond detection, such a problem seems to be able to bring more in-depth understanding of humor, including the effects of atomic and subtle edits and the recognization of some underlying patterns to machines as well. This might be helpful to build tools that could give advice on text editing or even text generating. Possible substitutes (based on POS tags or word similarities) might be automatically checked and those that tend to bring more humor could be given to real authors or even a text generator. Chat bots and translation AIs might also become more humorous based on this.

2 Problem Definition

Given the dataset described in next section, we already have measurements of the intensity of funniness. And based on these data, a model is expected to be built and trained so that it could be able to predict the average grade of funniness given the original and the edited headline.

3 Data

Humicroedit, a dataset introduced in (Hossain et al., 2019), is used for the training and testing of the problem. The data are collected news headlines, with specified words *micro-edit*ed to be funnier, where *micro-edit* is defined to be any of the replacements listed in Table 1.

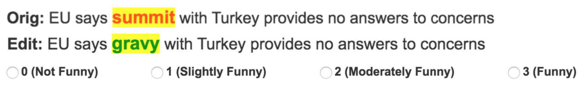
Replaced	Replacement
entity	noun
noun	noun
verb	verb

Table 1: Possible *micro-edit* replacements

And the funniness of edited headlines are manually graded by 5 judges based on the criterion as described in (Hossain et al., 2019).

Original Headline	Substitute	Avg Grade
Kushner to visit **Mexico** following latest Trump tirades	**therapist**	2.8
Hilllary Clinton Staffers Considered Campaign Slogan 'Because It's Her **Turn**'	**fault**	2.8
The Latest: BBC cuts **ties** with Myanmar TV station	**pies**	1.8
Oklahoma isn't **working**. Can anyone fix this failing American state?	**okay**	0.0
4 **soldiers** killed in Nagorno-Karabakh fighting: Officials	**rabbits**	0.0

Table 2: Sampled headlines in the dataset with the edits and average grades.



Orig: EU says **summit** with Turkey provides no answers to concerns

Edit: EU says **gravy** with Turkey provides no answers to concerns

○ 0 (Not Funny) ○ 1 (Slightly Funny) ○ 2 (Moderately Funny) ○ 3 (Funny)

Figure 1: The Grading of Headlines

The average of the 5 grades are used as the measurement of the funniness of the edited headline. Some sampled data are shown in Table 2.

The dataset used is already selected and preprocessed by *codalab*, which is split into train part and validation part. The train part contains 9652 rows of headlines, edits and corresponding mean grades. And the validation part contains 2419 rows of data.

A distribution of mean grades is visualized below.

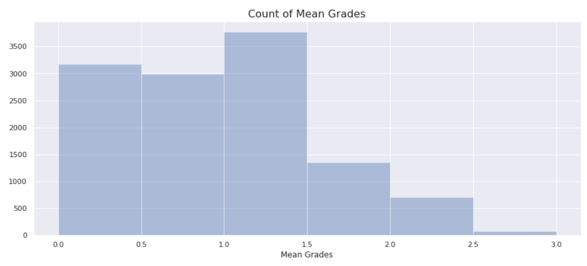


Figure 2: The distribution of mean grades

4 Related Work

After introducing the dataset, models are also built and tested to predict the funniness with only edited data in (Hossain et al., 2019). Using 80% of data for training and the rest 20% for testing, logistic regression, random forest, SVM, and a single-layer bidirectional LSTM (Long Short-Term Memory) model. And the single-layer bidirectional LSTM model is found to achieve the best performance in prediction.

There are some previous tasks raising similar problems of humor detection and funniness score prediction, for instance in the task Humor Analysis based on Human Annotation (HAHA) as described in (Castro et al., 2018), and Hashtag Wars (HW) as described in (Potash et al., 2017). What's different is that, both of the two previous tasks use raw tweets data without intentional edition for humor.

Despite some differences in the specific problems, we can still get insights from how the HAHA task and HW task are solved. For instance, Kernel Ridge Regression (KRR) is tested to be the best when predicting funniness scores in HAHA task by INGEOTEC team as described in (Ortiz-Bejar et al., 2018). And in (Muñiz Cuza et al., 2018), a bidirectional LSTM model similar to the one in (Hossain et al., 2019) is created for prediction.

Bidirectional Encoder Representations from Transformers (BERT), as a pretrained implementation of bidirectional LSTM as described in (Devlin et al., 2018), can be applied to detect the humor, as shown in (Mao and Liu, 2019) on HAHA task. The pretrained language model can help it better understand the meaning.

But for this particular task, besides the whole sentence, we also want to consider features related to the edited words, e.g. the original and edited versions, the POS tags, and etc. These extra features may or may not help detect the changes of humor.

5 Methodology

5.1 Training with *Doc2Vec* Embedding

It is natural to encode sentences with *Doc2Vec* embeddings and then classifiers with them.

One way is to just replace the original word with the corresponding edited word and directly encode the tokenized edited sentences. And further, we can also use embedding of both original and edited headlines.

We can build a vocabulary and train the *Doc2Vec* encoding based on text in the training dataset to make an encoder transforming a sentence to vectors. After fine-tuning based on testing on models like Random Forsest Regressors, a vecter size of 300 and the learning rate of 0.002 is used.

With embedding obtained from the text, we can try to build classifiers based on different models. According to what is suggested in the related works, Random Forsest Regressors and Kernel ridge regressors are expected to have better performances than others.

For the Random Forsest Regressor, we can adjust the max depth and number of estimators and validate the trained model. And similarly we can test on other regressors like Kernel Ridge Regressors and SVM.

5.2 BERT model

BERT, as a powerful implementation of bidirectional LSTM, can also be used for regression with text input. Since data in **Humicroedit** dataset are all collected news headlines, the pretrained BERT languae model trained on wikipedia data is good enough for the task.

Similarly, we can use either the edited headlines or both edited and original ones as input for training. Tokenize the headlines and add padding and special tokens ”[CLS]” and ”[SEP]”, we get vectors with the length of 32. Using the vectors as input and mean grades as output, we can train a classifier based on the model. The original and edited headlines are also combined in the form “[CLS] $edited\_text$ [SEP] $original\_word$ to $edited\_word$ [SEP]”, where ”$edited\_text$ ”, ”$original\_word$ ”, and ”$edited\_word$ ” are replaced by corresponding text respectively.

6 Evaluation and Results

For this regression task, the Root Mean Squared Error (RMSE) on the overall test set is calculated to measure the differences between the prediction of the mean grades and the actual values.

6.1 Baseline

Two models are used as the baseline for comparison. One is a random model which always gives the mathematical expectation value based on the distribution of mean grades in the training dataset and the other one is an Support Vector Regressor trained with *Doc2Vec* embedding.

Since the mean grades are obtained from several integer grades from 0 to 3, there are actually limited number of mean grades in the dataset. Based on the distribution in the training dataset, the mathematical expectation is calculated to be approximately 0.9356. The validation dataset then gives an RMSE of around 0.5784 if all the predictions are set 0.9356.

For the SVR regressor, the *Doc2Vec* embedding of every edited headline is used as input, and the regularization parameter C corresponding to the l2 penalty $\frac{1}{C^2}$ is fine tuned. RMSE on validation dataset is shown as follows.

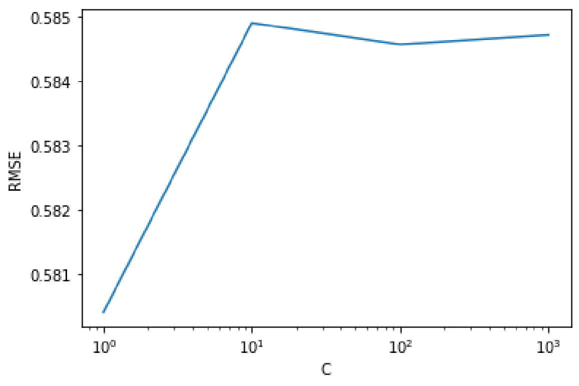


Figure 3: RMSE vs C

Even the best result (RMSE=0.5804) has an RMSE greater than 0.5784.

RMSE of 0.58 sounds a fairly good result, but the fact that the overall range is limited to 0 to 3 and that most observations are distributed densely from 0 to 1.5 makes it no better than a random guess based on the mathematical expectation.

6.2 Training with *Doc2Vec* Embedding

For the Random Forest Regressor trainedg with *Doc2Vec* embedding, when max depths of the model is fixed to 2, RMSE for models with different number of estimators are visualized as follows.

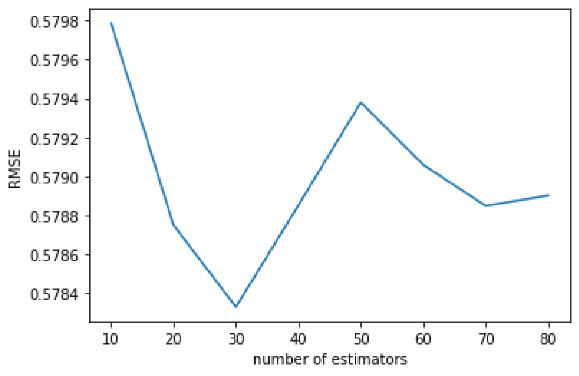


Figure 4: RMSE vs number of estimators (max_depth=2)

When number of estimators of the model is fixed to 10, RMSE for models with different number of max depths are visualized as follows.

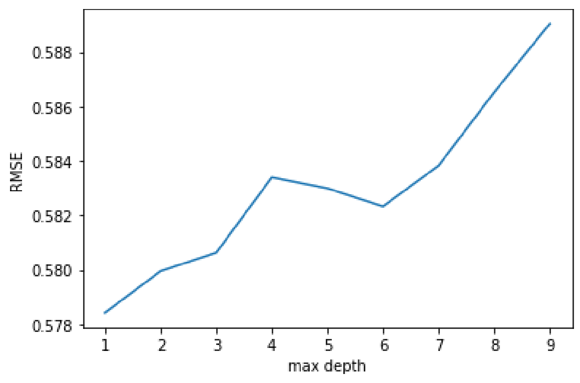


Figure 5: RMSE vs max depth (n_estimators=2)

As we can see, for a random forest regressor with max depth of 2 and number of estimators around 30, the Root Mean Squared Error (RMSE) score is around 0.578, which is approximately its best result.

When similarly training a Kernel Ridge regressor, we obtained the best result of RMSE around 0.58 when α is set to be 1.

When embeddings from both original and edited headlines are used, there isn't significant improvement on the RMSE score.

6.3 BERT model

After fine tuning the hyperparameters, a model with learning rate of 2e-5 and an adam epsilon of 1e-8 is found to have the best performance. The RMSE scores validated on development text after every epoch are shown as follows.

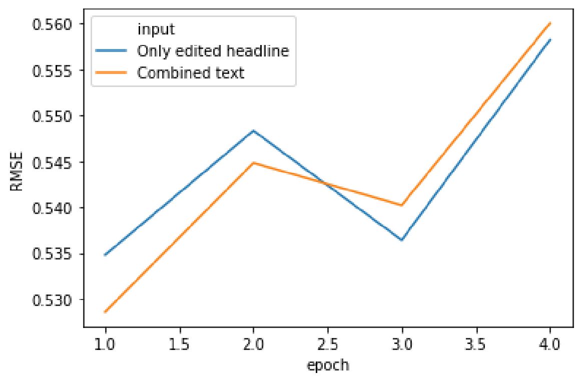


Figure 6: RMSE vs epochs for edidted headlines and BERT regressor

As we can see, BERT-based regressor can achieve RMSE of around 0.53, which is significantly better than the baselines. The best results are both obtained after one epoch, and they are quite close to each other in RMSE score, which means that adding original text does not significantly helps improve the performance.

7 Discussion

Similary to SVR, random forest and Kernel Ridge regressors trained with *Doc2Vec* embeddings achieves RMSE scores around 0.58, which does not seem to be better than the baselines. As for BERT-based models, an RMSE of around 0.53 is achieved, which is significantly better than baselines.

The BERT model seems to be able to detect some humor from given text. However, the performance is still not so satisfactory considering that the even the model which always predicts on the mathematical expectation value can achieve an RMSE of around 0.58.

The overall bad performances may be explanied by the limited range and dense distribution of actual mean grades in the dataset. For a model better at humor detection, we might need some better datasets with some larger grade range and more uniform distribution on grades.

8 Conclusion

In this project, we try to build a model for funniness detection by learning from text and grades in the **Humicroedit** dataset. Firstly, models like random forest and Kernel Ridge regresssors are tested given the *Doc2Vec* embedding of the text. These models achieve RMSE scores around 0.58, which are not better than a baseline model which always gives a prediction of the mathematical expectation values based on the distribution of the training dataset. Next, a BERT model, as an implementation of bidirectional LSTM model, is tested. Using the pretrained language model, the BERT regressor can achieve an RMSE of around 0.53, which is significantly better than the baselines. Code for this project is stored in https://github.com/theoliao1998/Natural-Language-Processing/tree/master/final_proj.

9 Other Things Tried

Considering that mean grades here are just means of 5 integer grades from 0 to 3, there are actually limited number of possible mean grades in the dataset. Labelling all the possible mean grades, we might treat it as a classification task. After training such a BERT-based classifier given the pre-trained BERT language model, the model gives an accuracy of around 0.05 on the validation dataset. However, for a model which randomly gives a label based on the distribution of mean grades in the training dataset, the accuracy on the validation set could reach as high as around 0.096. I failed to obtain a classifier better than a random guess (based on the distribution).

10 What could be done differently or next

When applying the BERT model, the regressor is directly trained from the given pretrained language model (trained with Wikipedia text). The language model might be further pretrained with news headlines which might help improve the

Also, we might try other methods to add extra feature to the BERT model. Adding features like POS of the edited words and similarity between the edited and original words might help.

11 Work Plan

In the initial few weeks, I studied the related works about humor and funniness detection and found that the bidirectional LSTM model is commonly applied and believed to give the best results. Then I further studied the details of the proposed bidirectional LSTM model as well as some other methods mentioned as planned.

Initially I planned to manually build a bidirectional LSTM model for regression. But I later found that the BERT model, as a powerful implementation of a bidirectional LSTM model with pretrained language model learning from wikipedia text, is easy and good enough to solve such a task. So I tried to dig deeper in the BERT model and managed to applied it to this regression task.

I also planned to add some extra features about the specific edited word for regression. However, I don't have a clear idea on what feature on earth should be obtained from the words and added for regression. Also, it could be complex to add other features to the implemented BERT model. So I finally decided to just try adding some text about the words before encoding. For the BERT model, the '[SEP]' token is added between the headline and the added text for separation.

Acknowledgments

References

Santiago Castro, Luis Chiruzzo, and Aiala Rosa. 2018. Overview of the haha task: Humor analysis based on human annotation at ibereval 2018 .

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* .

Nabil Hossain, John Krumm, and Michael Gamon. 2019. "President Vows to Cut ~~Taxes~~ Hair": Dataset and Analysis of Creative Text Editing for Humorous Headlines. *Proceedings of NAACL-HLT 2019* pages 133–142.

Jihang Mao and Wanli Liu. 2019. A bert-based approach for automatic humor detection and scoring. In *IberLEF@SEPLN*.

Carlos Muñiz Cuza, Reynier Bueno, Paolo Rosso, and José Pagola. 2018. UO-UPV: Deep Linguistic Humor Detection in Spanish Social Media.

José Ortiz-Bejar, Vladimir Salgado, Mario Graff, Daniela Moctezuma, Sabino Miranda-Jiménez, and Eric Tellez. 2018. INGEOTEC at Ibereval 2018 Task HaHa: TC and EvoMSA to Detect and Score Humor in Texts.

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, Vancouver, Canada, pages 49–57. https://doi.org/10.18653/v1/S17-2004.