UniTuebingenCL at SemEval-2020 Task 7: Humor Detection in News Headlines

Charlotte Sophie Ammer
University of Tuebingen, Germany
Department of Linguistics
charlottesophie72074@gmail.com

Lea Hannah Grüner
University of Tuebingen, Germany
Department of Linguistics
gauner.erle@gmail.com

Abstract

This paper describes the work done by the team UniTuebingenCL for the SemEval 2020 Task 7: "Assessing the Funniness of Edited News Headlines". We participated in both sub-tasks: sub-task A, given the original and the edited headline, predicting the mean funniness of the edited headline; and sub-task B, given the original headline and two edited versions, predicting which edited version is the funnier of the two. A Ridge Regression model using Elmo and Glove embeddings as well as Truncated Singular Value Decomposition was used as the final model. A long short term memory model recurrent network (LSTM) served as another approach for assessing the funniness of a headline. For the first sub-task, we experimented with the extraction of multiple features to achieve lower Root Mean Squared Error. The lowest Root Mean Squared Error achieved was 0.575 for sub-task A, and the highest Accuracy was 0.618 for sub-task B.

1 Introduction

With the rise of social media and social platforms such as Twitter and Facebook, the automatic interpretation of shorter amounts of text becomes increasingly more significant. The task of "reading between the lines" is so far often constricted to detecting offensive language or slurs. However, detecting funniness in written content also motivates research due to several reasons. Discovering humorous content can help understand text on a deeper level and get its intention. It not only helps comprehending the author, but also the readers of the text and why they favor it. Humor can be viewed as a natural language phenomenon and therefore linguistically interesting, but it also proves a lot of potential for modern psychology. Using humor can be a way to gain intimacy, reduce stress, and can improve feelings and overall wellbeing (Crawford and Caltabiano, 2011). It is often linked to soft skills like character strength and can therefore also be an asset for human resources, resulting in a change of hiring processes.

Humor is a communication skill that requires high levels of world knowledge and discourse insight. It is often provoked by an unexpected change of events, and has a lot to do with language usage and ambiguity. When used appropriately, humor can enrich human communication and serve as a tool to establish relationships. Humor has a sociological function and can therefore also be abused. Humor recognition can help detect such abusive humor, but also it may improve human-computer interaction. The task of automatic humor recognition is quite challenging because for a human to create or detect humor, world knowledge plays an important role. This is especially difficult in tasks where evaluation takes place without context dialogue to consider. Humor can also come in many variations, such as humor using anecdotes, self-deprecation, vulgarity and many more. With machine learning becoming more popular, it is a very interesting and contemporary unsolved task to assess how machines "learn" humor, especially in short text. The task of classifying to which extent a text is considered funny can be useful for fine-graining solutions for humor recognition problems.

Given a data set of English News Headlines and a corresponding dataset of the same, micro-edited headlines, the task of the competition was to estimate the funniness of the edited versions. The corresponding paper can be found under (Hossain et al., 2019). The headlines were single sentences taken from News. They were micro-edited, which means that one word was exchanged with the intention to make the sentence humorous. Sub-task A aimed at classifying the edited News Headline on a scale from 0 to 3, given the original Headline as a reference. In the provided data sets, the edits were classified by five human judges, and individual and mean scores were given. The goal was to predict the mean funniness of the headline. For example, given the original headline *"Recent Scandals Highlight Trump 's Chaotic Management Style."*, the system had to rank the funniness of the edited version: *"Recent Scandals Highlight Trump 's Chaotic Fashion Style."*.

Given two edited versions of the same original headline, sub-task B asked to predict which one of the edited versions is funnier. In the example above, a second edited version would be provided for comparison. For both sentences, the grades and mean grades were provided in the data sets. We participated in sub-task A and sub-task B.

The goal of this paper is to describe our work done in the SemEval 2020 Task 7. For task information, please consult the paper released by the authors of this task (Hossain et al., 2020).

2 Background

Computational research in the field of humor detection has had quite some history and is rooted in language theories. To dismantle humor, societal norms and structures have to be taken into account. The variety of the task allows for different analyses depending on the type of humor and the expression form. However, the detection of humor in text has been widely restricted to bigger amounts of text and shorter content has only recently been put into focus. Taylor and Marlack focused on one specific type of joke, namely wordplays. They aimed to extract structural patterns of jokes (Taylor and Mazlack, 2004). Two years later, Purandare and Litman identified humorous speech by analyzing spoken comedic conversations, taking into account gender and speaker effect on Humor-Prosody (Purandare and Litman, 2006). Since humor plays such a big role in societal structures, different theories were developed to explain the structure that lies underneath a good joke. The Incongruity Theory states that for a statement to be humorous, two different interpretations have to be present. Raskin 1984 used it as a basis for his Semantic-Script Theory of Verbal Humor that states that a text has to be compatible with two different scripts, such that the two scripts overlap partly or fully and are opposite (Raskin, 1984). The superiority theory (Hobbes, 1840) suggests that humor stems from a feeling of superiority to others or previous events. Minsky supports the relief theory, that states humor is a way to express taboos and societal unacceptable thoughts (Minsky, 1980).

Newspapers have been an important resource for decades and centuries. Headlines have been a big selling factor, and research has been done to unveil how headlines influence buyers (Winship and Allport, 1943). With the rise of social media, news have spread all over the Internet. With anonymity and accessibility, there come some risks such as abuse of the web for propaganda, racism and other negative topics. On the other hand, the Internet is spreading humor, and with social platforms such as Twitter rising, short, humorous text paragraphs become popular. When detecting humor in news, the headline often gives the seriousness away. It is therefore an important part of news analysis to also focus on the headline.

3 System Description

The core method used for our system was a L2 regularized linear model making use of the scikit-learn library provided in Python (Pedregosa et al., 2011). Input sentences were modified to lower-case, numeric-only text and saved for easy location of the edited word. Both original and edited version of the input were encoded using Tf-Idf weighting found in the scikit-learn library. Using two kinds of embeddings for the input sentence and adding the edit distance we aimed to facilitate humor recognition. An alternative approach was created by using an LSTM model using the Keras library provided in Python (Chollet and others, 2015). We constructed a twin LSTM architecture to be able to process the two versions of the input sentence simultaneously and create a merged output. Our code can be viewed following this link: https://github.com/leahannah/uniTuebingenCL .

5000 original headlines with two edited versions were provided in the data set. They were split randomly into a training set (64 %), a development set (16 %) and a test set containing the remaining twenty percent of the data. To use the provided data in the most efficient way, we modified it to fit our needs. We removed non-alphanumeric characters with a regular expression (Van Rossum and Drake, 2009). The input sentences were saved as lower-case strings with either the original or the edited word, which could be varied by changing a parameter. The numerical rankings by the judges provided in the data were saved as floats. We encoded sentences as n-grams by weighting them with tf-idf provided in the scikit-learn library. We chose to use Elmo embeddings as provided by AllenNLP (Peters et al., 2018). In order to use Elmo, a sentence array with clearly marked edited word was saved. For the LSTM approach the sentences were padded to have uniform length, determined by the maximum sentence length in the training set. Tuples containing the original and the edited word are saved for the embeddings. In sub-task B, we used a similar preprocessing structure as for sub-task A. We stored one ID for each sentence pair and for each of the sentences all other data was saved in the same manner as in task 1.

3.1 Sub-task A

Sub-task A was a regression task aiming to predict the mean funniness over all judged ratings for the edited newspaper headline, given the original and edited version. Root-Mean-Squared-Error was used to evaluate the model's predictions on the test set and development set.

Linear Model We set up a linear least squares model with L2 regularization. A non-regularized linear model performed worse due to the prediction of very high and low outliers. As a baseline we used the predictions of a linear model only trained on n-grams of the Tf-Idf Vectorizer. We normalized the predictions to make sure none is beyond the scale of zero to three. The parameters were tuned using a grid search (Pedregosa et al., 2011), a maximum document frequency of 0.95 and a minimum document frequency of 0.0005, which resulted in a large two to six gram feature space consisting of the training data as an output. The following features were found to be relevant to our process:

1. Truncated Singular Value Decomposition: helped reduce dimensionality and get compact and dense training vectors.

2. Edit distance between original and edited word: was appended to the feature vectors, since it had a slight effect on the RMSE.

3. 100d Glove word embeddings from Stanford NLP for the original and edited word: the Glove embeddings were appended to the feature vectors (Pennington et al., 2014). Other dimensions of Glove embeddings were tried (50d, 200d) but did not provide better results.

4. Elmo context embeddings (small) using FlairNLP for the original and edited word: Elmo embeddings were trained to context embed both the original and edited version of each sentence. Then, the context embeddings of the original and the edited word were appended to the feature vectors. We decided to use small Elmo embeddings due to limitations of computational resource and memory (Berlin, 2018).

LSTM Model A Twin LSTM architecture was set up using the Keras library in Python. All sentences were padded to the same length for this model, aiming to use both original and edited version of a sentence as an input to one part of the twin LSTM. Using the Glove 100d pre trained word embeddings from Stanford NLP, a Glove embedding layer was set up. We created an embedding matrix with a list of all unique words in the data, and encoded words in the embedding layer accordingly. We set up two twin LSTM structures, each consisting of an input layer, a Glove embedding layer, a dropout

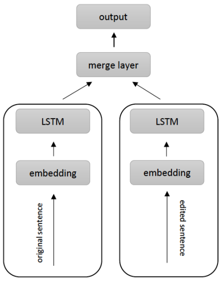


Figure 1: Twin LSTM

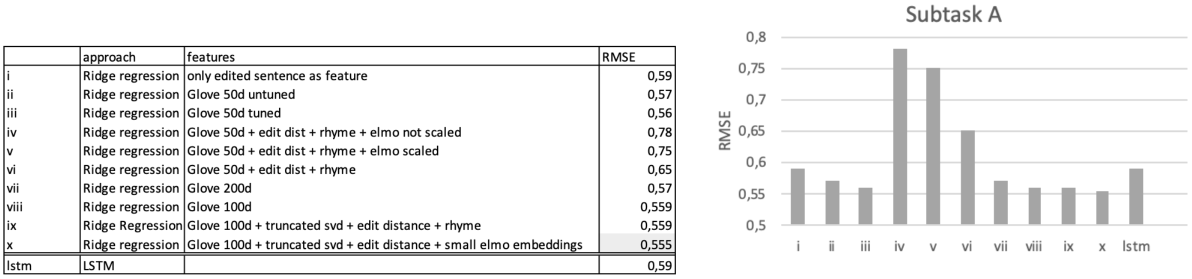
layer to prevent overfitting and an LSTM layer.
A visualization of the twin LSTM architecture is
given in Figure 1. To vary this approach, we also trained the small Elmo embeddings from FlairNLP on both original and edited sentences, and fed the output into a similar LSTM structure. The small Elmo embedding dataset consists of 13.6 million parameters and has an LSTM hidden size of 1024 and an output size of 128. The outputs of both models were concatenated in both approaches, and a Dense layer with 16 units was appended. In the case of the model consisting of Elmo embeddings, the outputs were flattened as well. The output layer had linear activation. We used Adam as optimizer and mean squared error as loss function, a validation split of 0.2, 100 epochs and a batch size of 16 after experiments with each of these parameters.

3.2 Sub-task B

Sub-task B aimed to predict the funnier of two versions of news headlines, where one word was edited. The accuracy of the predictions was used to evaluate the system. For example, the system was given two sentences that differed in one word, such as:
“5 things Trump did while you weren’t looking : Week 6” and the edited versions
“5 things eyelids did while you weren’t looking : Week 6” and *“5 things parents did while you weren’t looking : Week 6”*.
The system then had to compare the two edited sentences, having as reference the original version, and predict the funnier of the two. We used the same ridge regression model model as we did for sub-task A, described above. The parameters were taken from sub-task A as well since they were found to yield the best performance for this task, too. The two input sentences were split into two lists and fed into the Ridge Regression Model, which made a prediction about the funniness of each sentence. Then we compared the results of the prediction to determine the funnier of the two sentences. The system returned an integer, 0 if the sentences were equally funny, 1 if the first sentence was funnier and 2 when the second sentence was funnier.

4 Results

Figure 2 shows the results of our model tuning process for sub-task A. Trials i to x show the improvement of the ridge regression model over time. The last data point shows the twin LSTM structure in comparison. From the yielded scores we derived that the Glove 100d embeddings were the best dimension of embeddings for our purpose. Trial x lead to the best results, and was therefore used for the competitions test set. It outperformed the twin LSTM structure and became our main approach.



	approach	features	RMSE
i	Ridge regression	only edited sentence as feature	0,59
ii	Ridge regression	Glove 50d untuned	0,57
iii	Ridge regression	Glove 50d tuned	0,56
iv	Ridge regression	Glove 50d + edit dist + rhyme + elmo not scaled	0,78
v	Ridge regression	Glove 50d + edit dist + rhyme + elmo scaled	0,75
vi	Ridge regression	Glove 50d + edit dist + rhyme	0,65
vii	Ridge regression	Glove 200d	0,57
viii	Ridge regression	Glove 100d	0,559
ix	Ridge Regression	Glove 100d + truncated svd + edit distance + rhyme	0,559
x	Ridge regression	Glove 100d + truncated svd + edit distance + small elmo embeddings	0,555
lstm	LSTM		0,59

Figure 2: Ridge Regression Model tuning and LSTM comparison

In the evaluation phase, we submitted predictions by both systems, the twin LSTM structure and the L2 linear model. With the Linear Model, we achieved a RMSE of **0.539** and with the LSTM Model a RMSE of **0.575**. Therefore, the linear model predicted more accurately than the LSTM network. This left us on ranking place 18 of 49 participants in the competition.
For sub-task B, we adjusted the Linear Model from sub-task A to fit the new task structure. The parameters

of the Model were not changed. An accuracy of 0.618 lead us to the twenty-second place of the ranking. The results show that both tasks had their difficulties. Leaning mostly on semantic features to solve the task as seen in Figure 2, iv-vi, did not lead to the best results. Instead, a mixture of fine-tuning and pretrained embeddings have been shown to be efficient. Both sub-tasks, predicting a score of funniness and predicting the funnier of two sentences, were difficult to approach. The scores were provided by five judges and were their subjective assessment. We assume that this influences the difficulty of the task, since objective reasoning cannot be directly adopted as features for a Model. In sub-task A, we improved the RMSE a lot by experimenting with different parameters and features. For sub-task B, the same set-up did apparently not work quite as well.

5 Conclusion and future work

In this work, we presented a linear model and a neural network to solve the task of classifying the funniness of an one-line headline given a micro-edit (one semantic entity exchanged by another). We found some interesting approaches that might actually enable computers to improve their learning of humor. Throughout the competition, the task of predicting funniness given human judgement scores seemed overall challenging for most participants. The threshold of 0.5 RMSE score was surpassed by very few of us and only minimally. Since the topic is gaining popularity just recently, we hope to discover ways of improvement in future work.

References

Humboldt University Berlin. 2018. FlairNLP Elmo Embeddings: Documentation.

François Chollet et al. 2015. Keras. `https://keras.io`.

Shelley A Crawford and Nerina J Caltabiano. 2011. Promoting emotional well-being through the use of humour. *The Journal of Positive Psychology*, 6(3):237–252.

Thomas Hobbes. 1840. Human nature in english works.

Nabil Hossain, John Krumm, and Michael Gamon. 2019. "president vows to cut <taxes> hair": Dataset and analysis of creative text editing for humorous headlines. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 133–142, Minneapolis, Minnesota, June. Association for Computational Linguistics.

Nabil Hossain, John Krumm, Michael Gamon, and Henry Kautz. 2020. Semeval-2020 Task 7: Assessing humor in edited news headlines. In *Proceedings of International Workshop on Semantic Evaluation (SemEval-2020)*, Barcelona, Spain.

Marvin Minsky. 1980. Jokes and the logic of the cognitive unconscious. In *Cognitive constraints on communication*, pages 175–200. Springer.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proc. of NAACL*.

Amruta Purandare and Diane Litman. 2006. Humor: Prosody analysis and automatic recognition for f* r* i* e* n* d* s. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 208–215.

V Raskin. 1984. *Semantic Mechanisms of Humor*, volume 24. Springer Science & Business Media.

Julia M Taylor and Lawrence J Mazlack. 2004. Computationally recognizing wordplay in jokes. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 26.

Guido Van Rossum and Fred L. Drake. 2009. *Python 3 Reference Manual*. CreateSpace, Scotts Valley, CA.

Elizabeth C Winship and Gordon W Allport. 1943. Do rosy headlines sell newspapers? *Public Opinion Quarterly*, 7(2):205–210.