HumorAAC at SemEval-2020 Task 7: Assessing the Funniness of Edited News Headlines through regression and Trump mentions

Anna-Katharina Dick **Charlotte Weirich** **Alla Kutkina**
University Tübingen
{anna-katharina.dick, charlotte.weirich, alla.kutkina}@student.uni-tuebingen.de

Abstract

In this paper we describe the HumorAAC system, our contribution to the Semeval-2020 Humor Assessment task. We essentially use three different features that are passed into a ridge regression to determine a funniness score for an edited news headline: statistical, count-based features, semantic features and contextual information. For deciding which one of two given edited headlines is funnier, we additionally use scoring information and logistic regression. Our work was mostly concentrated on investigating features, rather than improving prediction based on pre-trained language models. The resulting system is task-specific, lightweight and performs above the majority baseline. Our experiments indicate that features related to socio-cultural context, in our case mentions of Donald Trump, generally perform better than context-independent features like headline length.

1 Introduction

Humor is an essential part of natural everyday human communication. A huge range of research was devoted to the nature of humor in various fields such as psychology, linguistics and philosophy resulting in various theories such as incongruity-resolution theory, non-cooperative theory, superiority theory, release theories, semantic script theory and general theory of verbal humor (Attardo, 2008). None of these theories could give a complete and comprehensive explanation of humor phenomena so far. For the field of computational linguistics humor is being quite a difficulty as well. Because of its complexity and inherent subjectivity, the development of automatic humor recognition and assessment poses a great challenge in Computational Linguistics, and therefore is a popular subject in various shared task competitions (Chiruzzo et al., 2019; Hossain et al., 2020) and independent research (Reyes et al., 2012; Mihalcea and Strapparava, 2005; Zhang and Liu, 2014; Yang et al., 2015). This challenge is related to the meaning extraction task in the domain of sentiment analysis since a great deal of naturally produced text contains humor in one form or another. The system described in this paper was created for the humor assessment task within the International Workshop on Semantic Evaluation SemEval-2020 (Hossain et al., 2020). This task is aiming to develop a system that automatically evaluates the humorousness of the edited headlines, made funnier with short one-word edits, and evaluated by human annotators. We believe, that this task can be an important step on a way to the development of better speech recognition systems and to the understanding of humor as a phenomenon in general.

The presented system is based on ridge regression and neural networks. In essence we pass a set of count-based, semantic, and context-oriented features into the ridge regression algorithm. The main discovery of our research is the importance of task- and data-specific features on the performance of a humor-detection system. Due to the social, context-dependent nature of humor, the addition of two features pertaining to Donald Trump, originally added only for experimental purposes, significantly improved our score. Therefore we can conclude that social context features like this are even more relevant to the perceived funniness than some other statistical features, for example those proposed by

Proceedings of the 14th International Workshop on Semantic Evaluation, pages 1019–1025
Barcelona, Spain (Online), December 12, 2020.

Hossain et al. (2020). This finding can help to improve humor detection systems by taking social context into account. An automatic way of detecting a social context is a perspective topic for future research.

2 System Overview

2.1 Tasks & Data

The development of the system was based on the Humicroedit dataset (Hossain et al., 2019): Regular English news headlines sourced from the r/worldnews and r/politics subreddit[1] are paired with versions of the same headlines that contain simple, one-word replacement edits designed to make them funny. These were subsequently scored by 5 judges that assigned a score between 0 and 3 to each edited headline (0 = not funny, 1 = slightly funny, 2 = funny, 3 = very funny). The data is stored as shown in Table 1. For Task 2 the data was stored as two headlines entries as in Task 1 with a common ID and the addition of a label detecting which of the headlines is funnier (0 = both headlines are equally funny, 1 = the first headline is funnier, 2 = the second headline is funnier). In total, the dataset contains 15,095 edited headlines. Both the editing and judging was crowdsourced. The expected output of the systems in this competition for task 1 is a predicted mean grade funniness score. The output for task 2 is a prediction of a label for the funnier headline.

ID	original headline	edited word	annotator scores	mean grade
14530	France is ‘ hunting down its citizens who joined <Isis\>’ without trial in Iraq	twins	10000	0.2
13034	“Pentagon claims 2,000 % increase in Russian trolls after <Syria\>strikes . What does that mean ?”	bowling	33110	1.6
8731	Iceland PM Calls Snap Vote as Pedophile Furor Crashes <Coalition\>	party	22100	1.0
3404	“President Trump ’s first year <anniversary\>report card , with grades from A + to F”	Kindergarten	33333	3.0
6164	Trump was told weeks ago that Flynn misled <Vice\>President .	school	00000	0.0

Table 1: Data example for task 1

2.2 Task 1

During our experiments with feature engineering the following features were tested:

- length of the headline
- absolute position of edited word within the headline
- relative position of edited word within the headline
- TF-IDF vectors
- word distance between original and edited word
- semantic similarity between original and edited word
- Trump mention
- Trump-hair co-occurrence
- Output of two alternative neural network architectures:
  - a sequential feed-forward model with a recurring layer trained on the edited headlines to predict the funniness score
  - a similar model using the edited word in the headline as auxiliary input (Figure 1)

We based a few of the features we used in our system on possible features mentioned in Hossain et al. (2020), namely length of the headline and the position of the edited word within the headline. The motive behind these features was that a longer headline has more potential to be funny because there are more possibilities to make edits, and that an edit in the later part of the headline could possibly contribute to the funniness following the setup-punchline theory of humor. We experimented with both the absolute and relative position of the word within the headline.

[1] www.reddit.com

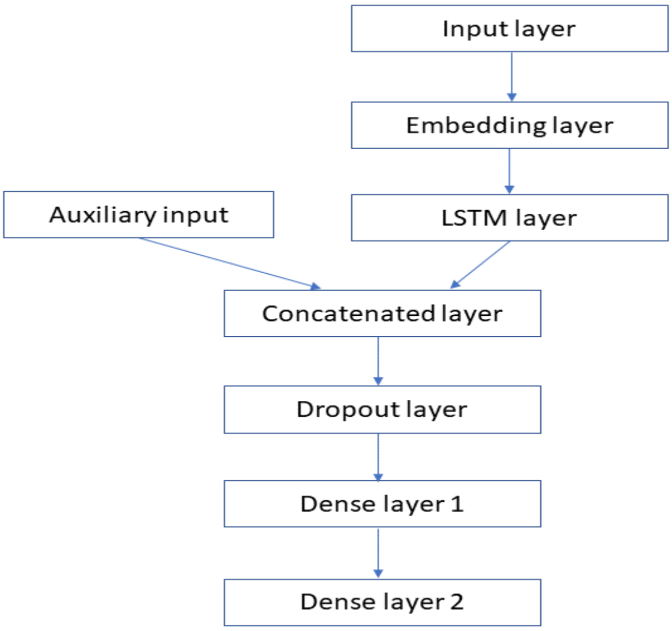

Figure 1: Neural network architecture

To catch humor related to just switching a few letters within the edited word but changing the meaning (e.g. “The Latest: BBC cuts ties with Myanmar TV station” replaced by “pies”), we added the feature of edit distance.

Another feature we experimented with is semantic similarity between the substituted word in the original headline and the edited word. We compared the lemmas from both words with the path similarity method. This method returns a score denoting the similarity of the two word’s senses in a hypernym/hyponym relation. The motivation behind this was to catch funniness resulting from seeing a word in a context that it does not fit into, as it is pretty likely that the substituted word will not fit into the context if it is very semantically different from the original word.

The most unique and experimental features in our system are two binary features we added after noticing a particular quirk within the dataset: Quite a lot of headlines feature President Trump and most of them are scored as funny, especially if there is some additional mention of hair. More than a third of the edited headlines in the trainset (35.5%) contained a reference to Trump. The likelihood of a headline mentioning Trump to be very funny (score of 2 or higher) is 9%, but when a headline contains both a reference to Trump and hair, the likelihood of it being scored as a 2 or higher is 45.3%. For reference, only 6.5% of all edited headlines in the trainset score this high.

We experimented with the integration of tf-idf vectors, comparing using the vectors for the original headlines, edited headlines, the edited word and the replacement word and even n-grams around the edit. Unfortunately we were not able to find a way to include these vectors in a productive way into our system that improved the score significantly enough to make up for the higher computational cost and long runtime. We suspect that this may be because tf-idf vectors work best on long texts and the headlines were pretty short. Furthermore the short style of the headlines that omits most function words may not have been the best fit for this method.

In order to integrate context information, we decided to build a neural network that uses pre-trained word representation vectors as input and predicts a funniness score. This prediction is also used as an additional feature for the final ridge regression. The first neural network we built and experimented with works on just the edited headlines: They are transformed into GloVe (Pennington et al., 2014) vectors and passed to an embedding layer, an LSTM layer with 10 units and tanh activation function, a dropout layer with dropout rate 0.2 and two consecutive dense layers with one unit and linear activation function each. The model was then trained for 50 epochs maximum and Early Stopping was used to prevent overfitting.

Since we wanted to not only model the funniness of the edited headline but also take the original headline into consideration, we built a second network that also uses the replaced words. This network has the same general architecture, input and output as the first, but uses the vector of the replaced word as additional input (see Figure 1). This auxiliary input is added to the network after the LSTM layer by adding a concatenated layer that concatenates the output of the LSTM layer with the auxiliary input.

2.3 Task 2

Our system for task 2 uses almost the same features as in task 1 but two different scoring alternatives. The first calculates a funniness score for both headlines like in task 1 and scores them according to the specifications for task 2. The second alternative uses this scoring as additional information, meaning it uses all the features from task 1 plus the calculated funniness scores between 0 and 3 for both headlines and the overall funnier headline based on these scores as features. A logistic regression is then fit on this feature set and the actual scores indicating which headline is funnier. The idea behind this is that the model can learn from its own predictions and balance out some of its assumptions that way, as well as allowing for additional tuning. The only feature that was dropped from the model in task 1 is the semantic similarity because it affected the runtime too much for how little improvement it provided.

3 Experimental setup

The competition's evaluation metric is the root-mean-square error (RMSE) for Task 1 and label accuracy for task 2. To evaluate our system, we both used a 80/20 train/test split on the train set as well as using the development set (both sets provided by task organizers). The final evaluation to determine our ranking was based on an additional test set. For all evaluation not through Codalab, building and tuning linear models, and feature extraction with TF-IDF we used the Python scikit-learn[2] library. For tokenizing, part-of-speech tagging and semantic similarity information (WordNet) as well as calculating the edit distance, NLTK[3] was used. As input for our neural network, headlines were preprocessed using Keras[4] and transformed with gloVe. We chose the pre-trained 100d vector variant sourced from Wikipedia and Gigaword 5 containing 6 Billion tokens [5]. The neural networks themselves are also implemented with Keras. We did not tune our neural networks when it comes to the number of layers or parameters but we used early stopping as a callback to prevent overfitting.

4 Results

Our system ranked 27th of 49 participating teams during the evaluation phase in task 1 with a RMSE score of 0.5645. Because of a bug in the tuning for task 2 we placed last in the evaluation phase although we did also score average in the development and post-evaluation phase. The overall best score our system reached was 0.5593 for task 1 and 58.60% accuracy for task 2. Interestingly, the scores for this competition were relatively close in range to each other: The majority of scores were less than 0.05 RMSE apart.

As shown in the table 1, the Trump and Trump-hair features show the most improvement on their own. This gives us interesting insights in how humor could work, and what can be necessary for humor detection task besides various language models: even encoded by language humor is not only linguistic, but also a complex socio-cultural phenomenon. Measuring the length of the headline did not improve the score much, this feature just barely surpassed the majority baseline on its own. This seems to support what Hossain et al. noted: Longer headlines might have more potential to be funny because there are more possibilities to place the micro-edit, but a short headline with economical word use places more focus

[2] https://scikit-learn.org/stable/index.html

[3] https://www.nltk.org

[4] https://keras.io/

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

Feature	development score	test score
random baseline (mean of 10 random predictions)	1.1839	1.1837
majority baseline	0.5784	0.5747
headline length	0.5783	0.5743
absolute position	0.5763	0.5738
relative position	0.5758	0.5742
semantic similarity	0.5784	0.5745
edit distance	0.5781	0.5745
Trump mention	0.5750	0.5698
Trump and hair mention	0.5769	0.5718
tfidf originals	0.6104	0.5980
tfidf originals with SVD	0.6087	0.5988
best features combined	**0.5715**	**0.5664**

Table 2: Feature scores on untuned Ridge Regression (used features in bold)

Feature	development score	test score
random prediction baseline	33%	33%
majority baseline	45.82%	43.55%
no regression, voting based on task 1	51.62%	47.02%
basic configuration without tuning	55.41%	50.02%
added tuning	56.69%	51.71%
added predictions and prediction of funnier headline	56.65%	52.28%
added tuning	58.60%	52.54%

Table 3: Accuracy scores for Task 2

on a funny edit (Hossain et al., 2019). We suspect that very short headlines are not likely to be funny because editors were too constrained to choose a good word to replace and very long headlines are either too complicated to be funny or editors were overwhelmed with choosing the best word to edit. The position of the edited word within the headline performed better and affirmed that an edit in the later part of the headline (modeled by the relative position) acts as a punchline and positively affects the perceived funniness. The absolute position in the headline describes a kind of hybrid between the aforementioned features and fittingly this feature's score lies between these two. When it comes to the neural networks, we found that simply concatenating the original and edited headline does not work well to model the funniness difference between them. On the other hand, adding the replaced word as additional input improved our model compared to our basic sequential model that only uses the edited headlines. The average RMSE score of predictions of the basic network (without regression) was around 0.6342 but the improved version reached 0.5712. For Task 2, using an additional logistic regression and adding the scoring using the predictions did improve the score a bit but also increased the runtime since the feature set is rather large.

5 Conclusions

To conclude our research we would like to reflect a bit more on a humor nature and humor recognition challenges. Humor is not only a linguistic phenomenon, but it can be represented by text and therefore provides a significant challenge for the NLP field. It is clear that humor cannot be accessed only by a language model by itself, no matter how good it is. Therefore the humor detection systems should be provided with some context not only in the sense of linguistic context but a broader, socio-cultural context. In our case such a context was provided by manual data exploration and implementation of corresponding features, but there are ways of doing it automatically. Just checking if a sentence contains a reference to Trump is not enough information to assess funniness of course, especially since his specific relevancy in culture might fade over the years. One of the possible ways to detect social-context related features would be named entity recognition in the dataset. Names can be crucially important to the task of obtaining social context, since they carry a lot of social-related information, being at the same time a references to certain common ground of readers. Thus in our example Trump denotes not just a person, but a whole social phenomenon related to him, including associations, attitudes and ideas. Therefore recognising, categorising and evaluating named entities can be helpful to detect socio-cultural context, being significant for the overall performance of a humor detection system.

References

Salvatore Attardo. 2008. Semantics and pragmatics of humor. *Language and Linguistics Compass*, 2(6):1203–1215.

Luis Chiruzzo, S Castro, Mathias Etcheverry, Diego Garat, Juan José Prada, and Aiala Rosá. 2019. Overview of HAHA at IberLEF 2019: Humor analysis based on human annotation. In *Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2019). CEUR Workshop Proceedings, CEUR-WS, Bilbao, Spain (9 2019).*

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.

Rada Mihalcea and Carlo Strapparava. 2005. Making computers laugh: Investigations in automatic humor recognition. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 531–538. Association for Computational Linguistics.

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.

Antonio Reyes, Paolo Rosso, and Davide Buscaldi. 2012. From humor recognition to irony detection: The figurative language of social media. *Data & Knowledge Engineering*, 74:1–12.

Diyi Yang, Alon Lavie, Chris Dyer, and Eduard Hovy. 2015. Humor recognition and humor anchor extraction. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2367–2376.

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*, pages 889–898.

6 Appendix

Parameter	**tuning**
alpha	6.0
fit_intercept	True
normalize	False
copy_X	True
max_iter	None
tol	1e-3
solver	auto

Table 4: Tuning for ridge regression in task 1

Parameter	**tuning**
penalty	l2
dual	false
tol	1e-4
C	1.0
fit_intercept	True
intercept_scaling	True
class_weight	None
random_state	None
solver	lbfgs
max_iter	100
multi_class	auto
verbose	0
warm_start	False
n_jobs	None
l1_ratio	None

Table 5: Tuning for logistic regression in task 2