SemEval-2020 Task 7: Assessing Humor in Edited News Headlines - Subtask 2 (Funnier) with RoBERTa

Manuel Hettich
Deep Learning for Natural Language Processing – SS21 – Philipp Cimiano and Philipp Heinisch

August 29, 2021

1 Introduction

This project is a possible solution to the SemEval-2020 Task 7: Assessing Humor in Edited News Headlines - Subtask 2 (Funnier) with the transformer model RoBERTa (Liu et al., 2019) using an implementation in Python by Hugging Face (Wolf et al., 2020).

The SemEval-2020 Task 7 deals with short humorous edits to news headlines in order to determine if machines can understand which edits are necessary to make a text funny. For this purpose, the authors of this task created new publicly available dataset, called Humicroedit, which contains 15,095 edited news headlines and the humor ratings of five judges per headline (Hossain, Krumm, and Gamon, 2019). Most of the edits consist only of replacing a single word, which helps to find the tipping point at which a normal text becomes funny. They were created by crowdsourced editors using original headlines from Reddit and curated by the paper's authors. The judges assigned a numerical rating between 0 - 3 and the ground truth funniness of each data point is the mean of its respective ratings. This grading procedure possibly enables a system to generate multiple edits to a regular text and rank them automatically by their probable funniness. Subtask 2 of SemEval-2020 Task 7 concerns the prediction of the funnier of two edited headlines. We are given the original headline as well as two edited versions of the same headline to predict the funnier version of the two by using the classification accuracy as evaluation metric.

This task is a good first step to better understand what makes a regular text become funny by editing and possibly enable machines to create them automatically, rate jokes from professional comedians and personalize funny headlines to individual readers. The data also connects with classic theories of humor, such as incongruity, superiority and setup/punchline. The authors argue that computerized humor generation has not seen much progress while automatic humor recognition made some good advances over the last few years. They claim this is not surprising since the same principle applies to humans: they can easily recognize something funny but often fail to produce funny content themselves due to the complex nature of what humor entails (in-depth world-knowledge, common sense and relationships of objects acress different layers of understanding). This challenge is further increased by the shortness of news headlines and the small edits employed in this task.

In this paper, a possible solution to the given task is demonstrated with the pretrained language model (PLM) RoBERTa, which is based on Google's BERT model and was released in 2019. At first, related work is described by taking a closer look at another published solution to the task which differs from my own. Afterwards, the model I have used and the corresponding architecture as well as my chosen language model are introduced in more detail. Finally, my test results are compared to the related work and some conclusions are discussed in order to possibly improve my approach in the future.

2 Related work

The team UniTuebingenCL (Charlotte Sophie Ammer, Lea Hannah Grüner) submitted their work for both sub-tasks of the shared task 7 at the 14th International Workshop on Semantic Evaluations in

2020 (Ammer and Grüner, 2020). They employed a ridge regression model using Elmo and Glove embeddings as well as Truncated Singular Value Decomposition. The original and edited versions of each headline were encoded using Tf-Idf weighting and they took the edit distance between the two different embeddings into account. In addition, they also used a long short term memory model recurrent network (LSTM) with the Keras library in a twin architecture to process the different headline versions simultaneously resulting in a merged output. In terms of pre-processing, the authors removed non-alphanumeric characters, turned all characters into lower-case and encoded sentences as n-grams by using their Tf-Idf weights. Also the sentences were padded to achieve uniform lengths for the LSTM approach.

In order to solve subtask 2, the authors used the same ridge regression they used in sub-task 1 with the same parameters. The predictions of the model for each of the two headline versions were compared to determine the funnier version as the result for this sub-task. The ridge regression model for sub-task 1 used a normalization of the output to limit the predictions at a value between zero and three and their parameters were tuned using a grid search.

The author's approach landed them at positions 18/48 for sub-task 1 and 14/31 for sub-task 2 with an accuracy value of 0.6183 in the competition. They notice the advantage of their ridge regression model with Glove 100d embeddings for sub-task 1 over their alternative twin LSTM approach and also received a better ranking with their linear model in the evaluation phase. The authors describe the challenges of both sub-tasks and conclude that a mixture of fine-tuning and pretrained embeddings yielded the best results and that their experimenting with different parameters and features did not result in better predictions for subtask 2. Finally, they claim the threshold of 0.5 RMSE score for sub-task 1 was surpassed by only few of the competitors and the tasks seemed challenging for most participants.

3 Method

For this project I have chosen to use the RoBERTa model, which was proposed by Liu et al. in 2019 (Liu et al., 2019) and it is based on Google's BERT model released in 2018. I am using the Python implementation by Hugging Face (Wolf et al., 2020) and I selected this model because it demonstrated the best test results in previous benchmarks for the same task (Hossain, Krumm, Gamon, and Kautz, 2020), achieving a similar accuracy to the fourth best team in the SemEval-2020 competition.

The RoBERTa model has modified hyperparameters, removed the next-sentence pretraining objective from the original BERT model and it was trained on larger mini-batches and learning rates as well as on longer sequences and the masking pattern applied to the training data was changed dynamically. The authors of RoBERTa state that according to their research the BERT model was "significantly undertrained" and that their improvements can "match or exceed the performance of every mode published after it" (Liu et al., 2019). They achieved similar results to other state-of-the-art models in different NLP benchmark datasets like GLUE, RACE and SQuAD. Interestingly, RoBERTa was also trained on the CC-News corpus, which contains over 60 million English news articles and it might explain some of RoBERTa's advantages over other approaches for the news headlines data in this task.

In general, transformer models like RoBERTa are deep learning models based on neural networks which employ the concept of attention by putting different weights on different parts of the input (Vaswani et al., 2017). Besides natural language processing (NLP), they are also used in computer vision programs and they have reduced training times due to parallelization compared to recurrent neural networks (RNNs) which process input data in order. Instead, transformers identify the context of each word in the sentence and thereby unmask their meaning. This approach enables researchers to use parallelization, since different parts of the input data can be processed at the same time and nowadays, transformers have become the preferred model for NLP tasks (Wolf et al., 2020).

In order to tokenize each data entry in the Humicroedit dataset, I am pre-processing it by generating two sentences per entry, one with the original word and one with the edited word included. Both sentences are stringed together and tokenized as a pair of sequences using the pre-trained RoBERTa tokenizer by HuggingFace (class transformers.RobertaTokenizer, version "roberta-base"). A pair of

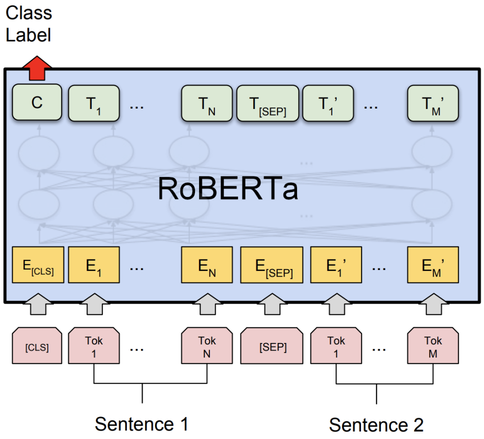


Figure 1: RoBERTa processing a sequence pair as input with special tokens displayed.

sequences "A" and "B" are concatenated using the language model's special tokens in the following way: `<s> A </s></s> B </s>`. Afterwards, the sequence pairs are encoded into input IDs which are used for training the model or calculating its prediction for the resulting label (see figure 1).

For training the language model, I am using a similar approach to the author's of UniTuebingenCL (Ammer and Grüner, 2020) in that I am training it first on the Humicroedit dataset for subtask 1 with a RMSE metric for 10 epochs and then use it as a single-shot inference for predicting a grade for both versions of a news headline in the test dataset for subtask 2. Finally, both predictions for a data entry are compared and the higher score determines which version is predicted as being the funnier one. The accuracy metric is used to determine how well the model has been trained for this task. The model itself is a specific version of RoBERTa, which is especially well suited for sequence classification / regression tasks like this one. It contains a linear layer on top of the final pooled output with 768 hidden states, 12 attention heads and 12 layers.

4 Evaluation

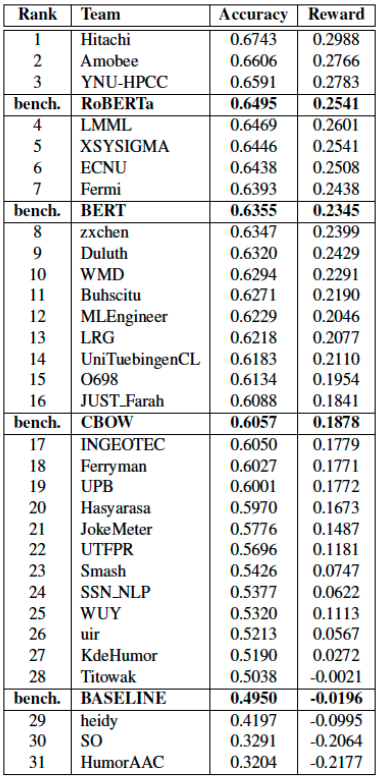
My final final test runs result in an accuracy value of 0.4533 for subtask 2, which is significantly worse than the model UniTuebingenCL (Ammer and Grüner, 2020) with an accuracy of 0.618 for the same task. In the official competition this result would have placed me at ranking 29/32 according to the published results and benchmarks (see figure 2).

It is also interesting to note that most predictions of my fine-tuned language model are very similar with a predicted funniness grade at around 1.0 per comparison of original vs. edited headline as a sequence pair. This result is similar to the author's findings of the original dataset (Hossain, Krumm, and Gamon, 2019).

Additionally, fine-tuning the language model took a very long time and I was only able to shorten this process by a certain factor by using Google's Colab features. However, limited computational resources in the free tier of their service restricted me in my efforts to benchmark different hyperparameter settings and only allowed me to run few training rounds.

5 Conclusion

Due to limited computational resource without a GPU, I am unfortunately unable to train my chosen language model for a significant amount of time on my personal computer hardware, which resulted in rather poor results in my test runs. It would be interesting to see if a longer training duration or an extended search for better hyperparameter settings would result in significantly improvements of my test results.



Rank	Team	Accuracy	Reward
1	Hitachi	0.6743	0.2988
2	Amobee	0.6606	0.2766
3	YNU-HPCC	0.6591	0.2783
bench.	**RoBERTa**	**0.6495**	**0.2541**
4	LMML	0.6469	0.2601
5	XSYSIGMA	0.6446	0.2541
6	ECNU	0.6438	0.2508
7	Fermi	0.6393	0.2438
bench.	**BERT**	**0.6355**	**0.2345**
8	zxchen	0.6347	0.2399
9	Duluth	0.6320	0.2429
10	WMD	0.6294	0.2291
11	Buhscitu	0.6271	0.2190
12	MLEngineer	0.6229	0.2046
13	LRG	0.6218	0.2077
14	UniTuebingenCL	0.6183	0.2110
15	O698	0.6134	0.1954
16	JUST_Farah	0.6088	0.1841
bench.	**CBOW**	**0.6057**	**0.1878**
17	INGEOTEC	0.6050	0.1779
18	Ferryman	0.6027	0.1771
19	UPB	0.6001	0.1772
20	Hasyarasa	0.5970	0.1673
21	JokeMeter	0.5776	0.1487
22	UTFPR	0.5696	0.1181
23	Smash	0.5426	0.0747
24	SSN_NLP	0.5377	0.0622
25	WUY	0.5320	0.1113
26	uir	0.5213	0.0567
27	KdeHumor	0.5190	0.0272
28	Titowak	0.5038	-0.0021
bench.	**BASELINE**	**0.4950**	**-0.0196**
29	heidy	0.4197	-0.0995
30	SO	0.3291	-0.2064
31	HumorAAC	0.3204	-0.2177

Figure 2: Official results and benchmarks for Subtask 2.

In general, I believe RoBERTa and other BERT-based language models are well-suited for this kind of classification task and further improvements in the pre-training strategies are destined to result in better outcomes for everyone without the need for significant computational resources.

References

Ammer, Charlotte and Lea Grüner (Dec. 2020). "UniTuebingenCL at SemEval-2020 Task 7: Humor Detection in News Headlines". In: *Proceedings of the Fourteenth Workshop on Semantic Evaluation.* Barcelona (online): International Committee for Computational Linguistics, pp. 1060–1065. URL: `https://aclanthology.org/2020.semeval-1.139`.

Hossain, Nabil, John Krumm, and Michael Gamon (June 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).* Minneapolis, Minnesota: Association for Computational Linguistics, pp. 133–142. DOI: `10.18653/v1/N19-1012`. URL: `https://aclanthology.org/N19-1012`.

Hossain, Nabil, John Krumm, Michael Gamon, and Henry A. Kautz (2020). "SemEval-2020 Task 7: Assessing Humor in Edited News Headlines". In: *CoRR* abs/2008.00304. arXiv: `2008.00304`. URL: `https://arxiv.org/abs/2008.00304`.

Liu, Yinhan et al. (2019). "RoBERTa: A Robustly Optimized BERT Pretraining Approach". In: *CoRR* abs/1907.11692. arXiv: `1907.11692`. URL: `http://arxiv.org/abs/1907.11692`.

Vaswani, Ashish et al. (2017). "Attention Is All You Need". In: *CoRR* abs/1706.03762. arXiv: `1706.03762`. URL: `http://arxiv.org/abs/1706.03762`.

Wolf, Thomas et al. (Oct. 2020). "Transformers: State-of-the-Art Natural Language Processing". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations.* Online: Association for Computational Linguistics, pp. 38–45. DOI: `10.18653/v1/2020.emnlp-demos.6`. URL: `https://aclanthology.org/2020.emnlp-demos.6`.