Joke Recognition Using Convolutional Neural Networks

Szymon Fonau
Felix Quinque

May 2019

1 Introduction

Humour is an important part of human language, whether for strengthening social bonds or improving mood. Humour has no precise definition due to its subjective nature but there exist many theories of humour. The lack of an exact understanding of what humour is makes it very hard to detect using conventional algorithms.

Being able to recognize and to generate humour is important as it can play a crucial role in human-computer interfacing. Humour can humanize natural language interfaces, it can make error messages less patronizing and repetitive and in general make the agent more pleasant and relatable [4]. In general users have been shown to find systems with humour more likeable and competent [5].

2 Previous Literature

Early studies of humour recognition were mostly based on linguistic features and usually focused on specific types of humour. For example Taylor and Marlack studies a specific part of humour, wordplays. They used an algorithm which was based on patterns and structures found in jokes [6]. But with the rise of artificial neural networks, the focus of methodology fell on them. A duo of researchers [3] used a recurrent neural network in order to detect humour in Yelp reviews. They also used a convolutional neural network and showed that a this type of network produces better results in humour recognition. Furthermore, convolutional neural networks have also been found to be better sentence encoders for humour detection [1]. In 2018 Chen and Soo wrote a paper on humour recognition using convolutional neural networks using multiple datasets [2]. The datasets seemed to have major domain differences which could explain their good performance.

We propose that the major differences in the domain of the datasets may cause a systematic error in the experiment. In order to investigate this, we will run an experiment using a convolutional neural network using a negative dataset of news headlines and a different dataset of quotes or rather proverbs/sayings, which usually contain more common language.

3 Methods

3.1 Data

In order to test whether the network can distinguish between humorous and non-humorous sentences we need a positive (contains humour) and a negative dataset (does not contain humour). In our experiment we used 3 different datasets, for our positive data we used the short jokes dataset avail-

able via Kaggle which was also used previously by Chen and Soo. As for the negative datasets, we used the 'a million headlines' dataset and the 'quotes' dataset. Both negative datasets were compared to the jokes dataset separately in order to find whether there is a difference in how challenging the classification task is for the network. This can be done because the 'a million headlines' dataset consists of only news headlines. Such sentences contain vastly different language and thus are really easy to tell apart from jokes, for example: "Ibuprofen is my favorite headache medicine that also sounds like a reggae professor." and "turkey closes border to iraq evacuates embassy" have wildly different lengths and word use and are therefore much easier to distinguish. But if you take a sentence from the 'quote' dataset: "God gives me hope that there is something greater than us, something better and bigger than the here and now, that can help us live." You can see that the language use and length are much closer to the joke and thus making them harder to tell apart

3.2 Processing

The words were converted into 200-dimensional vectors using the pretrained GloVe embeddings. To feed the network arrays of equal length, a padding consisting of null vectors was added. If one of the words in the sentence could not converted using the GloVe embeddings, it was also represented with a null vector. Finally, the resulting three dimensional vector (size: number of sentences; maximal joke length; 200 [Word2Vec dimensions]) was fed into a CNN. Said CNN was created using Tensorflow and Keras and consists of seven layers. Firstly, a one dimensional convolutional layer iterates through the sentence, secondly, a max pooling layer reduces the output dimensions of the previous layer. Finally, multiple Dropout and Dense layers are added to simplify the output down to the binary classification. As an activation function ReLU is used in all layers except for the last one where a softmax activation function is used to convert the outputs to probabilities. The amount of neurons per layer was optimized experimentally through trial and error.

4 Results

In this section, we will showcase the outputs of our network and its performance compared to the paper by Chen and Soo. In table 1 we can clearly see the difference in performances of the 2 different datasets and also compared to the previous research. One should note that our better results using the headline dataset might be due to a less rigorous headline selection so this does not mean our network is necessarily better. But the goal of this experiment was not to improve on the network as much as it was to investigate the effect of the dataset on the network performance.

Accuracy headlines vs jokes	Accuracy quotes vs jokes	Previous paper (headlines vs jokes)
0.938	0.763	0.897

Table 1: Comparison of accuracy in different runs

One should also note that all the graphs and results were computed us-

ing test data and should therefore not show inflated accuracy because of over-training.

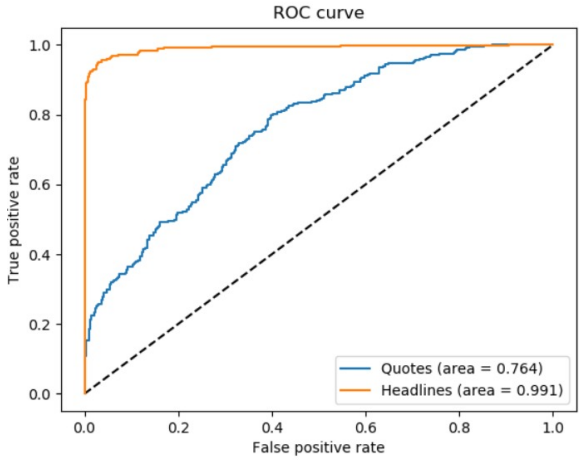

Figure 1: ROC curve comparison

Furthermore, Figure 1 shows how the performance of the network differs depending on what dataset is used for the negatives, confirming our hypothesis. It clearly shows that the classification between jokes and headlines is a much easier task, featuring an AUC (Area Under the Curve) of 0.991, whereas the classification between jokes and quotes only yields an AUC of 0.764. This proves the initial assumption that the experiments in previous papers contain a systematic error through the choice of datasets. The high accuracies found by previous papers (ca. 83%) is probably achieved because news headlines have very obvious identifying features (rare words, more condensed language, less colloquial terms etc.). We assume that our accuracy of 93.8% is due to a smaller size of the training and test datasets as well as a more complex network.

The findings of Figure 1 and Table 1 are also supported by Figure 2 and Figure 3. Figure 2 (jokes vs headlines) shows a nearly ideal histogram with barely any overlap in between the classes. Figure 3 (jokes vs quotes) shows a significant overlap of the classes, indicating more false predictions. Additionally, the typical peaks at 0 and 1 are far less pronounced, implying a more difficult classification task. Both histograms indicate that the respective networks hold predictive power.

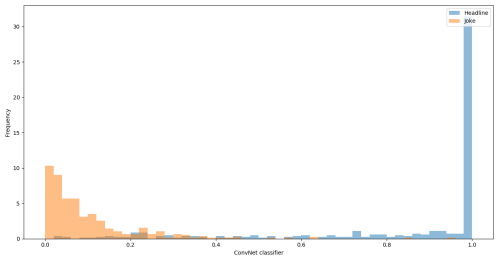

Figure 2: Histogram Jokes vs Headlines

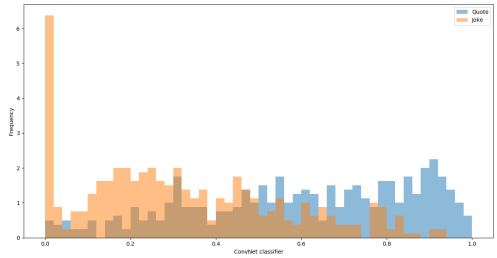

Figure 3: Histogram Jokes vs Quotes

5 Discussion

Our results showcase that there is a clear difference in performances depending on which dataset you use and from which domain it is. Chen and Soo obtained extremely good results using CNNs on their datasets because of the domains between positive and negative set were very different. Due to our less selective choice of headlines, their already high performance was increased by 5%. This, of course, does not change the fact that both their model and our model are an improvement on the state of the art, as most previous studies used the same or similar datasets.

Our results suggest a shift of focus of negative datasets used in creating humour detection networks towards a more diverse range of negative examples. Such that multiple non-humorous domains can be taken into consideration, making future models more diverse. If only headlines are used as a negative it cannot be tested whether a network identifies the jokes (as 'no normal text') or the headlines (as 'no normal text'). Therefore, it is necessary to conduct further studies on joke classification.

References

[1] Dario Bertero and Pascale Fung. A long short-term memory framework for predicting humor in dialogues. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 130–135, 2016.

[2] Peng-Yu Chen and Von-Wun Soo. Humor recognition using deep learning. Association for Computational Linguistics.

[3] Luke de Oliveira, ICME Stanford, and Alfredo Láinez Rodrigo. Humor detection in yelp reviews. 2017.

[4] Christian Hempelmann, Victor Raskin, and Katrina E Triezenberg. Computer, tell me a joke... but please make it funny: Computational humor with ontological semantics. In *FLAIRS Conference*, volume 13, pages 746–751.

[5] John Morkes, Hadyn K. Kernal, and Clifford Nass. Humor in task-oriented computer-mediated communication and human-computer interaction. In *CHI 98 Conference Summary on Human Factors in Computing Systems*, CHI '98, pages 215–216, New York, NY, USA, 1998. ACM.

[6] Mazlack L. J. Taylor, J. M. *Computationally recognizing wordplay in jokes.* In Proceedings of the Annual Meeting of the Cognitive Science Society (Vol. 26, No. 26), 2004.