Comparison of Classification- and Regression-Based Approaches for Humor Ranking on a #HashtagWars Twitter Dataset

Luka Čupić, Vinko Kašljević, Ivan Smoković

University of Zagreb, Faculty of Electrical Engineering and Computing
Unska 3, 10000 Zagreb, Croatia
{luka.cupic, vinko.kasljevic, ivan.smokovic}@fer.hr

Abstract

The comparison of classification- and regression-based methods in the context of natural language processing is a rather uncommon topic of contemporary work. While there is a certain intuition behind using either of these two approaches based on a particular problem, there might be some additional information that could be obtained by using the other approach. In this work, we attempt a regression-based approach for the problem of humor ranking to compare regression and classification. We hypothesize that by training a model to predict the *measure* of humor, the model could hypothetically understand this measure and predict it, potentially providing high-quality results. By trying this approach on several models of different architectures and complexities, we conclude that although regression shows some interesting results, classification works better in most cases.

1. Introduction

Humor can be dated, at least, to thousands of years ago – the ancient Romans, for example, had a sense of humor similar to ours, showing obvious examples of irony and sarcasm (Harvey, no date). Humor is treated as a characteristic human trait that arises naturally (for most) and is generally considered straightforward (excluding some extreme variants of sarcasm and such). In contrast, computers cannot understand humor the way humans do for obvious reasons.

In this work, we study the effects of using a regression-based approach for the problem of ranking humor on a collection of tweets. The tweets we use represent users' replies to hashtags from Comedy Central's @midnight show, which features a recurring #HashtagWars game. The game host presents a certain hashtag to which the contestants reply with witty responses. For example, some real responses for the #PrisonBooks hashtag are *100 Years of Solitary*, *Hijacker's Guide to the Galaxy*, and *Slaughterhouse Five... Three With Good Behavior*. Arguably, this kind of humor may be quite difficult for a model to learn, making the problem more interesting.

In this paper, we investigate the similarities and differences between classification and regression in terms of the model's output. An intuitive (and natural) approach (as has been done in most previous work) would be to represent the humor ranking problem in terms of classification. However, because of the nature of the used #HashtagWars dataset (described in detail in Section 3.), we hypothesize that such a task might be more appropriate for a regression-based approach. We also hypothesize that doing so could enable the model, instead of simply learning the correct class labels (as is the case with classification), to be able to predict the *humor measure* for each of the tweets. Working with a continuous scale rather than with discrete labels would introduce inter-class measures, which could potentially improve the model's accuracy. Additionally, in the context of our approach, we are actually solving a proxy problem: by learning the real-valued similarities between existing tweets, we hope to be able to apply these results to solving our original problem of ranking tweets. In the rest of this work, we present our approach for undertaking the described challenge of humor ranking.

2. Related Work

There have been different approaches to solving the humor ranking problem from the SemEval's competition. For instance, one of the competing teams, (Baziotis et al., 2017), used a Siamese architecture with bidirectional Long Short-Term Memory (LSTM) networks (Bromley et al., 1994); this approach won them 2^{nd} place on the competition, while their later improvement achieved state-of-the-art results on the #HashtagWars dataset. Another approach, which was the winning model at the competition, is an ensemble of a character-based convolutional neural network and an XGBoost (gradient boosting) model (Donahue et al., 2017).

Apart from the SemEval's competition, there have also been notable attempts at humor detection and ranking. Prior to SemEval's competition, most work on humor detection focused on detecting whether something *was* or *wasn't* humor. In other words, the majority of attempts were focused on building a binary classifier. Post competition, the contestants (as well as many others) have been inspired to perform humor ranking on a larger scale, thereby producing more information about the relationship between pairs of humor units (in this example, humorous tweets). However, a great many of them had approached the humor ranking as a classification problem, which is rather reasonable (Ortega-Bueno et al., 2018; Cattle and Ma, 2018; Donahue et al., 2017; Baziotis et al., 2017). Our interest, however, was to see how this could be posed as a regression problem, and whether the results would potentially outperform the classification-based approaches.

3. Dataset

The dataset used in this paper is provided as-is on the SemEval 2017 competition website and consists of training,

evaluation, and trial data.[1] We use these sets for training, testing, and validation respectively. The dataset contains 112 Twitter hashtags in total, with each of them having, on average, 114 associated tweets. Our training, testing, and validation sets consist of 101, 6, and 5 hashtags, respectively. In terms of the number of tweets, there are 11,321 tweets in our training set, 749 in our test set, and 660 in our validation set.

Each of the tweets represents a witty response to a specific hashtag given by a viewer of the #HashtagWars game. The tweets are annotated as follows: *0* means that the tweet *did not* reach the top 10; *1* means that the tweet *is* among the top ten tweets (but not the best one), while *2* means that the tweet has been chosen as the best one, i.e. the winning tweet for a given hashtag. The inherent natural ordering of these instance labels (*0*, *1*, and *2*) was the main reasoning behind our approach: instead of simply predicting the classes, the regression-based model might be able to recognize this ordering and potentially provide us with something more useful, such as a humor metric.

3.1. Pre-processing

For classification-based approaches, tweets were first grouped into pairs containing the texts of both tweets. Each pair consists of two tweets from the same hashtag, but of different humor ranking. For instance, a tweet that belongs in the "top 10" ranking could be paired with a tweet that belongs to the "below 10" ranking, or with the "top 1" tweet. After pairing, each tweet pair is flipped with a probability of 0.5 and a label indicating which of the tweets has the higher ranking is added. The reasoning behind the flipping is to avoid having pairs that always have the higher ranked tweet on one side, which would be trivial for any model to learn and overfit to. The value of the label is "0" if the first tweet is ranked higher, or "1" if the second tweet in the pair is ranked higher. Our classification-based models take the pre-processed tweet pairs and try to predict the ranking labels. There are 103124, 6944, and 6145 pairs of tweets in the pre-processed training, testing, and validation sets, respectively.

For regression-based approaches, each tweet from the dataset was assigned a label denoting the ranking for its hashtag. A tweet ranked "top 1" would be assigned a label of value "2", a tweet ranked "top 10" would be assigned a label of value "1" and a tweet ranked "below 10" would be assigned a label of value "0". The classes are then balanced by duplicating tweets of classes that have fewer examples than the most numerous class. This is repeated until all classes have the same number of examples. After this, the labels of all tweets are normalized to a range between 0 and 1. Our regression-based models take the tweet text as input and try to predict the normalized label, which is a proxy value for the comedic value of the tweet. For regression, our training set consists of 30966 tweets, whereas the validation and the test sets are identical to those from the classification approach.

[1] http://alt.qcri.org/semeval2017/task6/index.php?id=data-and-tools

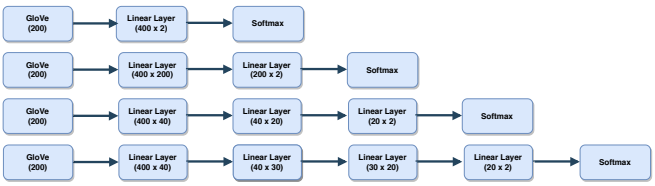

Figure 1: Architectures of classification-based models with GloVe embeddings. All layers imply a ReLU activation function.

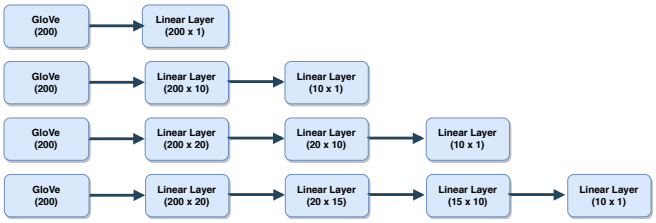

Figure 2: Architectures of regression-based models with GloVe embeddings. All layers (including the output) imply a ReLU activation function.

4. Approach

With our approach, we have tried a variety of available models. To start with, we defined the lowest accuracy benchmark (50%) with a random baseline model. We used several fully-connected models with GloVe Twitter embeddings (hereafter: GloVe embeddings). Furthermore, we used LSTM with both character and GloVe embeddings. We chose these models because they offer different layers of complexity and we wanted to see how model complexity affects accuracy for both regression and classification. In the rest of this section, we will describe these models in detail.

4.1. GloVe Embeddings

GloVe (Global Vectors) represents an unsupervised model for transforming words into vector-space (Pennington et al., 2014). We use GloVe (specifically, pre-trained word vectors for Twitter) as our main (most commonly used) feature extractor.[2]

In the context of GloVe embeddings, we use several models of increasingly higher complexities. For both classification and regression, we use GloVe embeddings of size 200 and a dropout value of 0.5. For each of the approaches, we constructed four fully-connected neural networks whose architectures can be seen in Figures 1 and 2. Figure 1 shows the models used for classification, while Figure 2 shows the models used for regression.

4.2. ELMo Embeddings

In addition to the GloVe embeddings, we also use ELMo. We use fully-connected models with ReLU activations and a dropout of 0.5. The layers' architectures (i.e. outputs and inputs) are similar to the architectures shown in Figures 1 and 2 but are left out for the sake of brevity.

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

Table 1: Comparison of classification and regression outputs for the used models and word embeddings. The first column represents the models used in this work; *GloVe* and *ELMo* refer to the GloVe and ELMo embeddings used on the models' inputs, whether on tokens or lemmas, and the final part represents the number of fully-connected hidden layers.

Model	**Classification**			**Regression**		
	ACC (Val)	ACC (Test)	NLL	ACC (Val)	ACC (Test)	MSE
Random Baseline	50.00	50.00	-	50.00	50.00	-
GloVe Token 1FF	62.64	50.74	0.6472	53.57	58.38	0.2268
GloVe Token 2FF	**64.97**	57.40	0.5051	56.35	55.45	0.1080
GloVe Token 3FF	64.02	52.14	**0.4953**	62.41	48.30	0.0849
GloVe Token 4FF	63.62	53.01	0.4994	**67.03**	53.49	**0.0640**
ELMo Lemma 1FF	61.87	62.08	0.6098	57.47	49.03	0.1865
ELMo Lemma 2FF	61.64	62.78	0.5977	56.37	56.43	0.1405
ELMo Lemma 3FF	62.06	62.42	0.6038	57.75	49.15	0.1178
Char LSTM	58.14	**68.21**	0.5908	55.87	**61.79**	0.1802
GloVe LSTM	64.17	61.18	0.5559	60.57	50.84	0.1245

4.3. Sequential Models

Finally, we also use two recurrent networks: character- and GloVe-based LSTM. The LSTM is used to produce embeddings, which we use as input to two fully-connected layers with ReLU as activation function and a single layer of dropout at the beginning. We get the embeddings by only using the last output of the LSTM. We used this model for both classification and regression; the only difference is that the number of weights for classification is twice as big as for regression. Additionally, classification also has the softmax function on the output.

4.4. Additional Approaches

Neutral Dataset Alongside the dataset described in the previous subsection, we have also included a dataset from another SemEval competition, designed for the task of performing sentiment analysis for a given collection of tweets.[3] This dataset contains annotated tweets distributed among three sentiment categories: positive, neutral, and negative. Since our main dataset and this one are contextually similar (in terms that they are both comprised of tweets), we added this dataset as a new category, which would represent neutral tweets. Our presumption with this decision was that the model could be trained to learn the "ground truth" for neutral tweets, so that it does not become particularly overfit to humorous tweets, thus enabling the model to become more robust for future predictions.

5. Experiments and Results

In this section, we describe the learning process for our models and present and discuss the obtained results.

As mentioned in Section 3., we use the training set (consisting of 11321 tweets distributed among 101 different hashtags) for training our models. We use batching, with each training instance using a batch-size of 5,000 samples. We also use the Adam optimizer with a learning rate of 0.001 and a weight decay of 0.001. The dropout value, as mentioned earlier, is fixed at 0.5. As for the initial parameters, all of them are randomly generated using a random number generator with a fixed seed value (100) to achieve reproducibility.

We use accuracy as the evaluation metric for both classification and regression approaches. For regression, as previously mentioned, we use training and validation sets identical to those from the classification approach. The accuracy is computed by first predicting the humor score and then using the argmax function to determine which of the tweets is more humorous, giving a label to the pair, as described in 3.1.

Table 1 shows our main results (in terms of the accuracy evaluation metric), which indicate that classification outperforms regression in the majority of cases. Further looking at the results, some more interesting figures can be seen from the table. Regression-based GloVe Token 3FF, for example, produces a 62.42% accuracy on the validation set, and a mere 48.30% on the test set. Another interesting example is with classification-based ELMo Lemma 1FF model, producing 62.08% on test set, unlike regression, which performs worse than the Random Baseline. The results are somewhat better for the Char LSTM model, but there is still a 6.42% difference between classification (68.21%) and regression (61.79%).

Unable to provide any tangible intuition for the achieved results, we provide some possible explanations: (i) regression-based approaches use considerably less data than classification-based ones; (ii) dimensionality of the input data for regression is half that of the dimensionality for classification; (iii) regression is trained on a different (proxy) problem compared to classification; and (iv) the dataset used for regression is somewhat perplexing. The last point needs some further elaboration: if the model is trying to learn a certain distribution on the train set, choosing a model from the second (test) distribution, and then measuring the accuracy on the third (validation) distribution, the results might not be satisfactory enough. In contrast to this, classification-based approaches might be able to overcome this problem due to having two tweets at the

[3] https://www.dropbox.com/s/byzr8yoda6bua1b

input.

We hypothesize that the problem might lie in the dataset. Looking at the results from Table 1, it is evident that in some cases, there is an inconsistency between accuracy on validation and test set; this was certainly an unwelcome surprise. GloVe Token regression model, for example, achieves poor accuracy on the validation set, but good accuracy on the test set.

Finally, when we take into account all previously mentioned issues with regression, it is not surprising that classification works better. Still, both regression and classification achieve some interesting figures.

5.1. Additional Results

Our additional approaches have, unfortunately, provided unsatisfactory results. The neutral dataset has been shown not to work as we expected, so we disregarded it in the early stages of our work.

6. Conclusion

In this work, we compared classification- and regression-based approaches in the context of humor ranking. We used a variety of models and word embeddings on the SemEval's dataset of humorous tweets. With the achieved results, we showed that while classification generally works better than regression for humor ranking, additional research could be performed on the subject of comparing regression and classification because we believe that there might be a particular subset of NLP problems in which the presented ideas and hypothesizes could be more applicable.

Acknowledgements

We would like to take this opportunity to express our endless gratitude to our dear colleague Ivan Smoković for leaving us to work in peace and quiet while he went to a rock concert in Vienna.

References

Christos Baziotis, Nikos Pelekis, and Christos Doulkeridis. 2017. DataStories at SemEval-2017 Task 6: Siamese LSTM with Attention for Humorous Text Comparison.

Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Sickinger, and Roopak Shah. 1994. Signature Verification using a "Siamese" Time Delay Neural Network.

Andrew Cattle and Xiaojuan Ma. 2018. Recognizing Humour using Word Associations and Humour Anchor Extraction.

David Donahue, Alexey Romanov, and Anna Rumshisky. 2017. HumorHawk at SemEval-2017 Task 6: Mixing Meaning and Sound for Humor Recognition.

Brian Harvey. no date. Graffiti from Pompeii. http://www.pompeiana.org/Resources/Ancient/Graffiti%20from%20Pompeii.htm.

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

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.