Humor Analysis Based on Human Annotation Challenge at IberLEF 2019: First-place Solution

Adilzhan Ismailov

Universitat Pompeu Fabra (Spain)
adilzhan.ismailov@upf.edu

Abstract. This paper describes the winning solution to the Humor Analysis based on Human Annotation (HAHA) task at IberLEF 2019. The main classification task is solved using an ensemble of a fine-tuned multilingual BERT (Bidirectional Encoder Representations from Transformers) model and a naive Bayes classifier.

Keywords: Natural Language Processing · Deep Neural Networks.

1 Introduction

Humor Analysis based on Human Annotation (HAHA) challenge [2] proposes two tasks: to classify tweets in Spanish as humorous or not, and rate how funny they are on a given scale. This paper describes the winning solution for both of these tasks.

The main classification task is solved using an ensemble of a fine-tuned multilingual BERT (Bidirectional Encoder Representations from Transformers [3]) model and a naive Bayes classifier. The solution achieves the F1 score of 0.821 with the second-place score of 0.816.

The regression task is also solved by fine-tuning a multilingual BERT model. The final submission is a weighted average of the regression BERT model and a LightGBM model (`https://github.com/microsoft/LightGBM`) estimated on TFIDF features. The solution achieves the root-mean-square error (RMSE) score of 0.736 with the second-place team achieving the RMSE of 0.746.

The rest of the paper is organised as follows. Section 2 describes the challenge and Section 3 describes the solutions for each of the tasks. Section 4 concludes.

2 Task Description

The challenge asks to classify tweets in Spanish as humorous or not, and rate how funny they are on a scale from one (not humorous) to five. The dataset [1]

is a corpus of crowd-annotated Spanish-language tweets split into a train and a test set. The train set consists of 24000 tweets out of which 38.6% are considered humorous with an average rating of 2.05. The test set comprises 6000 tweets for which only text is given. There are two tasks: Humour detection and Funniness score prediction:

Humour Detection: the goal is to classify tweets into jokes (intended humour by the author) and not jokes. The performance is measured using F1 score.

Funniness Score Prediction: the goal is to predict a funniness score (average of crowd-sourced ranks) for a tweet supposing it is a joke. The performance is measured using root-mean-square error.

3 Solution Description

For every model the final predictions were obtained by averaging predictions from each fold of a five-fold cross-validation schemes, in other words by averaging predictions from five models each estimated using 80% of the data. The validation scores reported below are F1 scores and RMSEs calculated on out-of-fold predictions for the full train set.

3.1 Classification task

The results are summarised in Table 1. The baseline provided by the organisers considers a tweet a joke randomly with a probability of 0.5.

The main model used in the final solution is fine-tuned multilingual cased BERT model (12-layer, 768-hidden, 12-heads, 110M parameters) [3] that supports 104 languages including Spanish. We use PyTorch implementation by HuggingFace[1] that also provides pretrained weights and vocabulary, and build on top of that. We tokenize the text by basic tokenization followed by WordPiece tokenization, following the original implementation and do not apply any preprocessing to the text.

For the classification task we use binary cross-entropy loss and one-cycle learning rate schedule [4]. Given the small dataset and large model capacity overfitting is a major issue, and to combat that we follow [4] and apply differential learning rates across the layers, with the maximum learning rate for the first layer set at a half of the maximum learning rate for the last layer, the classification head (set at 2e-5). The batch size used is 32 and we train the model for each fold for four epochs, and use the checkpoint for the epoch with the best validation F1 score for test-set predictions. The differential learning rates and one-cycle learning-rate schedule are implemented using FastAI library[2]. This base model achieves the score of 0.818 on cross-validation and 0.807 on the test set.

[1] `https://github.com/huggingface/pytorch-pretrained-BERT`
[2] `https://docs.fast.ai/`

To further reduce overfitting we fine-tune the pretrained weights by using unsupervised learning and text data from both train and test sets. The idea comes from the ULMFiT paper [4], where unsupervised fine-tuning on the text from the same domain as the target task before classification step significantly improved results. However while in [4] the unsupervised task is predicting the next word here we use the tasks used for training the original BERT language model: combination of masked language modelling and next sentence prediction loss, again by modifying HuggingFace library implementation to this task. We fine-tune the language model to the domain for ten epochs and use the obtained weights in place of the original pretrained weights for the classification task and repeat the steps above. This model achieves 0.829 on cross validation and 0.815 score on the test set, a significant improvement over the base model.

We then average these predictions with predictions from a naive Bayes model following [5] with a logistic regression[3] estimated on uni- and bi-grams features picked using TFIDF.

The model alone achieves only 0.771 on cross-validation, however an ensemble of the BERT model above and the naive Bayes model with weights 0.72 and 0.28 respectively scores 0.833 on cross validation and 0.821 on the leaderboard. The optimal weights are derived from cross-validation. This ensemble was the final solution for the classification task.

Table 1. Classification task results comparison

Submission	F1	Precision	Recall	Accuracy
BERT+LM tuning+NB-Log	0.821	0.791	0.852	0.855
2nd place solution	0.816	0.802	0.831	0.854
BERT+LM tuning	0.815	0.761	0.878	0.845
3rd place solution	0.810	0.782	0.839	0.846
BERT	0.807	0.775	0.842	0.843
Baseline	0.440	0.394	0.497	0.505

3.2 Regression task

The results are summarised in Table 2. Here the baseline is a constant prediction of 3 assigned to all items in the test. For the regression task we also ensemble two models: a fine-tuned BERT model and a LightGBM model based on gradient-boosted trees.

The only difference between the BERT model for the regression task and the BERT model for the classification task is the loss function - for former we use the mean-squared loss given that the challenge metric is RMSE. This model achieves RMSE of 0.726 on cross-validation and 0.746 on the test set.

[3] Based on the code from `https://www.kaggle.com/jhoward/nb-svm-strong-linear-baseline`

The second model is a LightGBM model estimated on the same features as the naive Bayes model above. The loss function is also mean-squared error and we set bagging and feature fraction parameters to 0.7 to add regularisation. This model scores 0.795 on cross validation.

For the final solution we average predictions from these two models linearly with weight of 0.71 assigned to the prediction from the BERT model and 0.29 - to the LightGBM model, with weights coming from the cross-validation. This ensemble scores 0.712 on cross-validation and 0.736 on the test set.

Table 2. Regression task results comparison

Submission	RMSE
BERT+LM tuning +GBM	0.736
BERT+LM tuning	0.746
2nd place solution	0.746
3rd place solution	0.769
BERT	0.895
Baseline	2.455

4 Conclusion

This paper describes the winning solution for both classification and regression tasks of the Humor Analysis based on Human Annotation challenge at IberLEF 2019, which consists of an ensemble of a fine-tuned BERT model and a complementary model estimated on TFIDF features derived from uni- and bi-grams.

Firstly, we can see that a high score can be achieved by solely appropriately fine-tuning a BERT model. In fact that model alone would rank second in this competition, and with longer hyperparameter search or longer language model fine-tuning - possibly rank first.

Secondly, unsupervised learning can help to reduce overfitting and improve the score significantly when the size of the dataset is small.

And, finally, adding weaker but different and diverse models to the ensemble helps to boost the score as has been demonstrated above.

Further work can be done both in improving the quality of the predictions and studying how the models assign them. For instance it would be interesting to examine predictions from different layers of the neural network to study where does attention fall and what token combinations make the model decide whether a tweet is humorous or not.

References

1. Castro, S., Chiruzzo, L., Rosá, A., Garat, D., Moncecchi, G.: A Crowd-Annotated Spanish Corpus for Humor Analysis. Proceedings of the Sixth International Workshop on Natural Language Processing for Social Media, 7–11 (2018)
2. Chiruzzo, L., Castro, S., Etcheverry, M., Garat, D., Prada, J., Rosá, A.: Overview of HAHA at IberLEF 2019: Humor Analysis based on Human Annotation. Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2019). CEUR Workshop Proceedings (2019)
3. Devlin, J., Chang, M. Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805. (2018)
4. Howard, J., Ruder, S.: Universal language model fine-tuning for text classification. arXiv preprint arXiv:1801.06146. (2018)
5. Wang S, Manning C. Baselines and bigrams: Simple, good sentiment and topic classification. Proceedings of the 50th annual meeting of the association for computational linguistics: Short papers-volume 2, 90-94. Association for Computational Linguistics. (2012)