FII FUNNY at SemEval-2021 Task 7: HaHackathon: Detecting and rating Humor and Offense

Mihai Samson
Alexandru Ioan Cuza University Iasi
mihai.samson@info.uaic.ro

Daniela Gifu
Alexandru Ioan Cuza University Iasi
daniela.gifu@info.uaic.ro

Abstract

The "HaHackathon: Detecting and Rating Humor and Offense" task at the SemEval 2021 competition focuses on detecting and rating the humor level in sentences, as well as the level of offensiveness contained in these texts with humoristic tones. In this paper we present an approach based on recent Deep Learning techniques by both trying to train the models based on the dataset solely and by trying to fine tune pretrained models on gigantic corpus.

1 Introduction

The figurative language of Social Media is one of the most challenging topics facing natural language processing (NLP). In this study, we refer at humor that requires a multidisciplinary approach for its detection (Dan Alexandru and Daniela Gˆıfu, 2020). Imagine, a viral topic on social media as elections (Gîfu, 2010) or a political crisis (Delmonte, Rodolfo and Tripodi, Rocco and Gîfu, Daniela, 2013). Social media users themselves introduce a specific language based on common practices (e.g., humor, irony), making their message analysis very challenging (Reyes, Antonio and Rosso, Paolo and Buscaldi, Davide, 2012). The legitimate research questions of this paper intend to answer: Is humor an insurmountable barrier for Artificial Intelligence (AI)? We propose an approach based on recent DL techniques for sentiment analysis (SA). Furthermore, we experimented with multiple types of DL architectures ranging from Convolutional Neural Networks (CNN) to Recurrent Neural Networks (RNN) which we tried to train using only the dataset provided by the SemEval-2021 Task 7 competition, but we also used pretrained Transformer architectures which we fine-tuned using the available data. The rest of the paper is organized as follows: section 2 describes the literature related to sentiment analysis and humor detection, section 3 presents the dataset and method of this study, section 4 summarizes the results of the conducted experiments, with discussions after experiments, followed by section 5 with the conclusions.

2 Related Work

This topic has attracted significant attention in recent years, evidenced by increasing number of workshops of the same competition (e.g., SemEval-2017 Task 6: HashtagWars: Learning a Sense of Humor or SemEval-2020 Task 7: Assessing Humor in Edited News Headlines). Such a competition is attractive, especially since the problem of labeled data is somewhat solved, considering the fact that the automatic humor recognition depends on these. For the binary task, as in this case, there are many computational models to solve it or to detect the humor intensity or humor dimension (Yang, Diyi and Lavie, Alon and Dyer, Chris and Hovy, Eduard, 2015) (Chen, Peng-Yu and Soo, Von-Wun, 2018). Thus, work on this topic was never followed by high results, as this problem is still almost subjective and text classification even for humans is very controversial and biased. Never the less, the task can be approached like any SA task for which most of the authors used LSTMs (Murthy, Dr and Allu, Shanmukha and Andhavarapu, Bhargavi and Bagadi, Mounika, 2020) or CNNs (Ouyang, Xi and Zhou, Pan and Li, Cheng Hua and Liu, Lijun, 2015). New approaches concentrate on using attention based methods (Vaswani, Ashish and Shazeer, Noam and Parmar, Niki and Uszkoreit, Jakob and Jones, Llion and Gomez, Aidan N and Kaiser, Lukasz and Polosukhin, Illia, 2017), in particular transformer architectures such as BERT (Devlin, Jacob and Chang, Ming-Wei and Lee, Kenton and Toutanova, Kristina, 2018), RoBERTa (Liu, Yinhan and Ott, Myle and Goyal, Naman and Du, Jingfei and Joshi, Mandar and Chen, Danqi and

Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021), pages 1226–1231
Bangkok, Thailand (online), August 5–6, 2021. ©2021 Association for Computational Linguistics

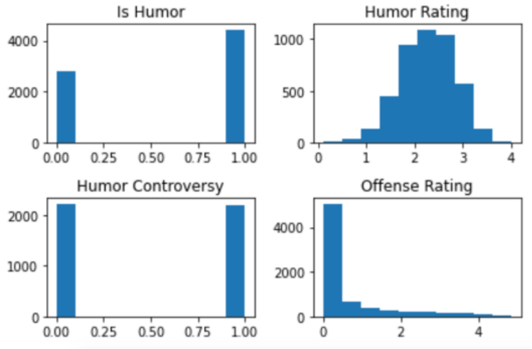

Figure 1: Data distributions by subtasks.

Levy, Omer and Lewis, Mike and Zettlemoyer, Luke and Stoyanov, Veselin, 2019), ALBERT (Lan, Zhenzhong and Chen, Mingda and Goodman, Sebastian and Gimpel, Kevin and Sharma, Piyush and Soricut, Radu, 2019), and VideoBERT (Sun, Chen and Myers, Austin and Vondrick, Carl and Murphy, Kevin and Schmid, Cordelia, 2019). These transformers are pre-trained on unlabeled data to be later fine-tuned for a variety of tasks. For this task we used the BERT architecture.

3 Dataset and Methods

This section contains details about the dataset built as part of SemEval-2021 Task 7 HaHackathon: Detecting and Rating Humor and Offense, which was the basis for solving the subtasks of this competition.

3.1 Dataset

The dataset (Meaney, J.A., and Wilson, Steven R. and Chiruzzo, Luis and Lopez, Adam and Magdy, Walid, 2021) consists of 8000 short texts that have four labels corresponding to the four subtasks of the competition. The first label is binary and determines if the text is humorous. If this label is 1, then it has associated two additional labels: one for the second task which is a number from 0 to 5 representing the average humorous score of the annotators and another denoting if the kind of humor is controversy, again a binary classification. Regardless of the previous 3 scores, the fourth is score from 0 to 5 denoting if the text is offensive. Because of these conditions we used the entire dataset only for the first and the fourth tasks, and for the second and third we only used the texts which were labeled to be humorous. Table 1 shows some examples from the dataset of SemEval-2021 Task 7.

Note that for the third task, the labels seemed to be randomly assigned, neither of our methods succeeding in obtaining a better performance than one we would obtain by flipping a coin. Because of this situation we will only present the results for the remaining three subtasks.

3.2 Method

In order to apply DL-based modeling techniques, we first need to embed the words in a vector form that the neural networks can work with. In order to achieve this, we used two methods depending on the architecture trained. For the models that we trained from scratch we used a tokenizer implemented in the Keras library [1] (TextVectorization) which performs the following steps exemplified on the sentence: “The quick brown fox jumps over the lazy dog.”

1. lowercasing and punctuation stripping; “the quick brown fox jumps over the lazy dog”

2. splitting the text into words; [“the”, “quick”, “brown”, “fox”, “jumps”, “over”, “the”, “lazy”, “dog”]

3. assembling tokes, assessing each token an index; “the”:1, “quick”:2, “brown”:3, “fox”:4, “jumps”:5, “over”:6, “lazy”:7, “dog”:8

4. transforming the text into a sequence of integers. [1, 2, 3, 4, 5, 6, 1, 7, 8]

After the tokenization, we mapped these indexes to the words from the GloVe embeddings which contains a vocabulary size of 400k words, more than enough for our task. We chose the predefined embedding size of 100 and standardized the texts to a length of 70. Figure 2 shows a histogram of the number of words after splitting the texts by space.

We also fine-tuned a pretrained BERT model which uses its own set of embeddings. In this case the huggingface [2] library which also provides the pretrained model offers the tools to embed the words into the necessary vectors specific to the selected model. After having the embeddings for either set of methods, we further describe the architectures used. We trained the following models from scratch:

[1] https://keras.io/

[2] https://huggingface.co/transformers/

id	text	is humor	humor rating	controversy	offense rating
35	Learn from the scars of others	0			0.05
119	What do you call a sad terrorist? A crisis	1	2.16	1	0.85
81	January is the Monday of months	1	2.43	0	0.00

Table 1: Examples of texts and their humor labels.

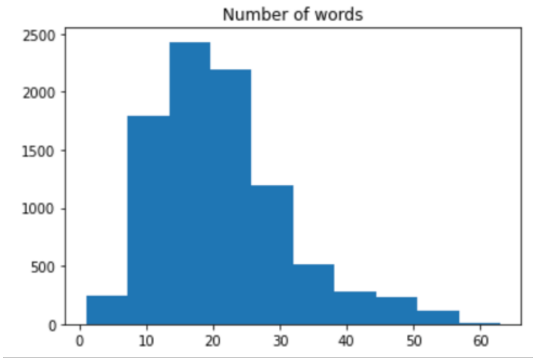

Figure 2: Histogram for the number of words in texts.

1. Convolutional Neural Networks (CNN). We created simple sequential models with 1D convolutions and max pooling layers followed by global average pooling and a few fully-connected layers. We haven't experimented with more advanced architectures like ResNet (Kaiming He and Xiangyu Zhang and Shaoqing Ren and Jian Sun, 2015) and Inception (Christian Szegedy and Wei Liu and Yangqing Jia and Pierre Sermanet and Scott E. Reed and Dragomir Anguelov and Dumitru Erhan and Vincent Vanhoucke and Andrew Rabinovich, 2014) because we found that even the simplest model drastically overfits the data.

2. Bidirectional LSTM and GRU cells. In order to take advantage of the natural structure of sentences we used the two most popular recurrent cells and we also wrapped them into a Bidirectional wrapper. Each cell was used in a separate architecture and we didn't stack more than one cell because we again found that it tends to overfit.

3. Transformer blocks. In order to take advantage of the recent developments in the NLP tasks, we created the encoder part of the original Attention is all you need paper. We used six encoding blocks followed by a global average pooling layer and a few fully-connected layers.

We employed a pretrained transformer model (BERT) from the huggingface library which was trained on cased datasets. We used the base model which has around 100 million parameters and used our dataset to only fine tune this model. Because only the first and fourth tasks used the entire dataset, we applied all these methods only on the first task, the classification between humorous and non-humorous. As it can be observed from Figure 1 the dataset is imbalanced, therefore we applied class-weight on the 0 class such that the loss from the two classes among the entire dataset will be equal. After experimenting with the hyperparameters(mainly with the dropout rate which finally we chose it to be 0.2) on this single task, the best configuration was used on the other three tasks, changing only the activation function for the regression tasks from Sigmoid to Rectified Linear Units (ReLU). For the classification tasks, the threshold for the prediction was 0.5 and for the regression ones the predictions from the neural networks were sealed to 5 in order to correspond to the competition's requirements. A diagram summarizing the system architecture can be seen in Figure 3.

The first step is to split randomly the available dataset into a training set and a validation set. We kept 10% of data for the validation set. Then, the next step is to preprocess the data. In order to have a good representation of the metrics on the validation set, we only computed the necessary tools for preprocessing on the training set and then applied these tools to the validation step. Most importantly, the vocabulary used in the models was selected based on the training set solely. For the pretrained BERT model, the tokenizer already contains a large vocabulary meant to cover most of the common words. After preprocessing the dataset, we moved forward to train the models with the corresponding embeddings. As mentioned, we did all the experiments on the first task and we experimented with the hyperparameters on this set. We did not employ a test set due to the small size of the dataset. Finally, we adapted the classification model trained

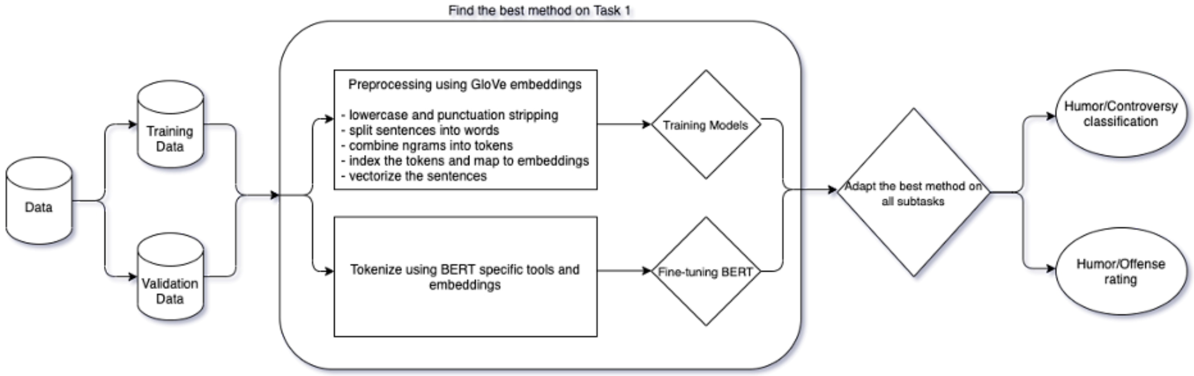
Find the best method on Task 1
Data
Training Data
Validation Data
Preprocessing using GloVe embeddings
- lowercase and punctuation stripping
- split sentences into words
- combine ngrams into tokens
- index the tokens and map to embeddings
- vectorize the sentences
Tokenize using BERT specific tools and embeddings
Training Models
Fine-tuning BERT
Adapt the best method on all subtasks
Humor/Controversy classification
Humor/Offense rating

Figure 3: System Architecture.

Task Name	RMSE	Position
Average Humor	0.5598	19 of 50
Average Offensiveness	0.4788	24 of 48

Table 2: Official scores for the regression tasks.

previously to the regression tasks and trained a separate model for each task using only the samples that had the corresponding labels. After all the four models have been trained, we made predictions on the evaluation set using the tokenizer for BERT and the corresponding model.

4 Results and Discussion

Below are presented the official results for all subtasks. For the classification tasks we also report the results in the post-evaluation phase and the ranking as of March 2021. In the official competition we accidentally performed a mistake for the Humor Detection task and reverted the labels that we submitted. That is why we obtained such a poor performance and this is the reason we also report these results as it better reflect the actual performance. We report Accuracy (Acc), F1-score (F1), and Root Mean Square Error (RMSE). The official results are summarized in Tables 2 and 3 and the results in the post-evaluation phase are presented in Table 4.

The results on our validation set for all the techniques are summarized in the graphics depicting the evolution of the metrics over the epochs (Figure 4). The test label refers to our validation set and not the official test set from the competition.

We can easily see that all the methods overfitted the training set. Techniques such as dropout and regularization have been applied, but we observed that they only delayed the moment when overfitting occurred and did not increase the performance on the validation set. The best accuracy on the validation set of the techniques used on the first task is presented in Table 5.

As it can be observed from the table, the best method turned out to be to fine tune a pretrained BERT model, therefore we used this technique on the rest of the tasks. Despite of its success we can still observe that the accuracy on the validation set (95.4%) differs from the one we obtained in the competition (92.2%). For approaches to SA on a small dataset, the best way is to fine tune pretrained models, rather than trying to train a model from scratch with the available data. The humor detection task turned out to be a very challenging one, fact that can be best expressed by the results on the second classification task where we assume that the very diverse interpretations of the annotators on what constitutes a controversy humor made the task impossible to solve with any DL model. But for more approachable tasks like the first classification task or the regression tasks, probably the more consensus among annotators made the task more tractable to an artificial intelligence model.

5 Conclusion

According with the legitimate question, we still consider that detecting someone's sense of humor is a difficult problem and identifying the level of offensiveness and irony is an even harder one. In this case, a figurative content could be consider irony, satire, joke, and sarcasm. As with humor, all these figures of speech depends on the listener or reader to be in on this context. This paper presents a system participating at SemEval 2021 Task 7 and tried to adapt existing Deep Learning techniques to the problem of humor detection. This approach indi-

Task Name	Acc	F-Score	Position
Humor Detection	7.8%	0.063	58 of 58
Humor Controversy	50.08%	0.4752	29 of 36

Table 3: Official scores for the classification tasks.

Task Name	Acc	F-Score	Position
Humor Detection	92.2%	0.9374	27 of 39
Humor Controversy	50.08%	0.4752	24 of 41

Table 4: Scores in the post-evaluation phase for the classification tasks as of March 2021.

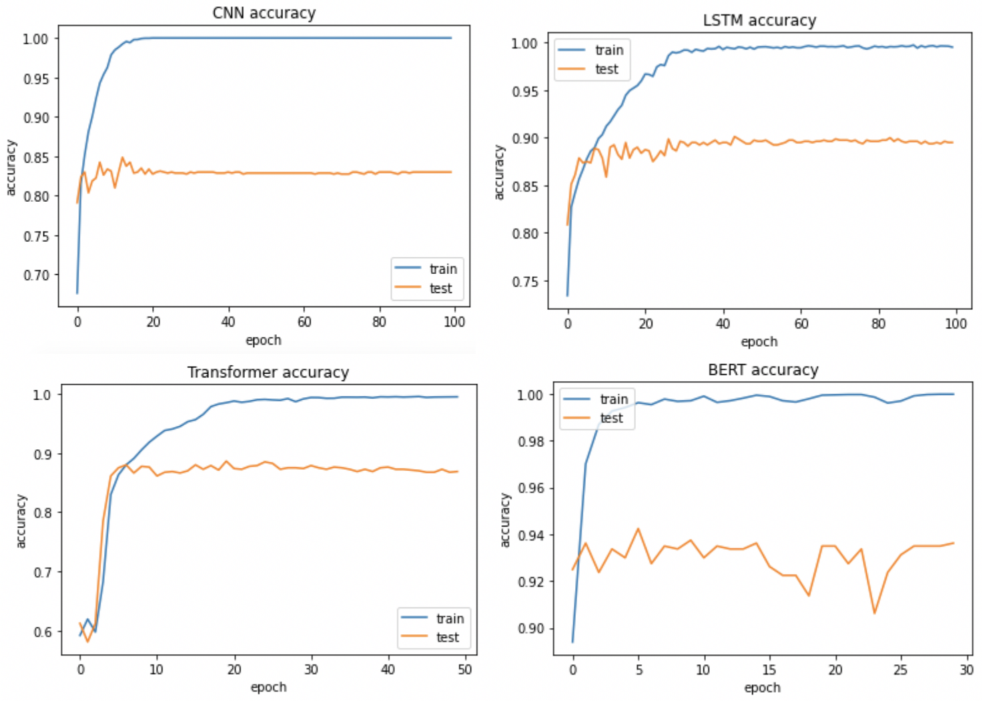

Figure 4: Evolution of accuracy .

Model	CNN	GRU	LSTM	Transformer	BERT
Validation Acc	84.9%	90.1%	90.1%	88.6%	95.4%

Table 5: The accuracy on the validation set for the 5 techniques on the Humor detection task.

cates promising results since they offer compelling results regarding the accuracy. We also found that we can get the best performance by adapting pre-trained models on other, bigger, datasets, indicating that the internal representation of language of the model acquired in other contexts can be extremely helpful in trying to identify the humor of a sentence. For further research ideas, we consider trying to use data augmentation techniques such as replacing words with synonyms, as well as using similar datasets in order to increase the dataset and the performance of the discussed methods.

References

Chen, Peng-Yu and Soo, Von-Wun. 2018. Humor Recognition Using Deep Learning. pages 113–117.

Christian Szegedy and Wei Liu and Yangqing Jia and Pierre Sermanet and Scott E. Reed and Dragomir Anguelov and Dumitru Erhan and Vincent Vanhoucke and Andrew Rabinovich. 2014. Going Deeper with Convolutions. *CoRR*, abs/1409.4842.

Dan Alexandru and Daniela Gˆıfu. 2020. Tracing Humor in Edited News Headlines. In *Ludic, Co-design and Tools Supporting Smart Learning Ecosystems and Smart Education*, pages 187–196. Springer Singapore.

Delmonte, Rodolfo and Tripodi, Rocco and Gîfu, Daniela. 2013. Opinion and Factivity Analysis of Italian Political Discourse. In *IIR*, pages 88–99. Citeseer.

Devlin, Jacob and Chang, Ming-Wei and Lee, Kenton and Toutanova, Kristina. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Kaiming He and Xiangyu Zhang and Shaoqing Ren and Jian Sun. 2015. Deep Residual Learning for Image Recognition. *CoRR*, abs/1512.03385.

Lan, Zhenzhong and Chen, Mingda and Goodman, Sebastian and Gimpel, Kevin and Sharma, Piyush and Soricut, Radu. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

Liu, Yinhan and Ott, Myle and Goyal, Naman and Du, Jingfei and Joshi, Mandar and Chen, Danqi and Levy, Omer and Lewis, Mike and Zettlemoyer, Luke and Stoyanov, Veselin. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Meaney, J.A., and Wilson, Steven R. and Chiruzzo, Luis and Lopez, Adam and Magdy, Walid. 2021. SemEval 2021 Task 7, HaHackathon, Detecting and Rating Humor and Offense. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*.

Murthy, Dr and Allu, Shanmukha and Andhavarapu, Bhargavi and Bagadi, Mounika. 2020. Text based sentiment analysis using LSTM. *Int. J. Eng. Res. Tech. Res*, 9(05).

Ouyang, Xi and Zhou, Pan and Li, Cheng Hua and Liu, Lijun. 2015. Sentiment analysis using convolutional neural network. In *2015 IEEE international conference on computer and information technology; ubiquitous computing and communications; dependable, autonomic and secure computing; pervasive intelligence and computing*, pages 2359–2364. IEEE.

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

Sun, Chen and Myers, Austin and Vondrick, Carl and Murphy, Kevin and Schmid, Cordelia. 2019. Videobert: A joint model for video and language representation learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7464–7473.

Vaswani, Ashish and Shazeer, Noam and Parmar, Niki and Uszkoreit, Jakob and Jones, Llion and Gomez, Aidan N and Kaiser, Lukasz and Polosukhin, Illia. 2017. Attention is all you need. *arXiv preprint arXiv:1706.03762*.

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