# ES-JUST at SemEval-2021 Task 7: Detecting and Rating Humor and Offensive Text Using Deep Learning

**Emran Al Bashabsheh**
Computer Science Department
Jordan University of
Science and Technology
Irbid, Jordan
emranalbashabsheh@gmail.com

**Sanaa Abu Alasal**
Computer Science Department
Jordan University of
Science and Technology
Irbid, Jordan
sanaasal11@gmail.com

## Abstract

This research presents the work of the team's ES-JUST at semEval-2021 task 7 for detecting and rating humor and offensive text using deep learning. The team evaluates several approaches (*i.e.BERT (Devlin et al., 2018), Roberta (Liu et al., 2019), XLM-Roberta (Conneau et al., 2019), and BERT embedding + Bi-LSTM*) that employ in four sub-tasks. The first sub-task deal with whether the text is humorous or not. The second sub-task is the degree of humor in the text if the first sub-task is humorous. The third sub-task represents the text is controversial or not if it is humorous. While in the last task is the degree of an offensive in the text. However, Roberta pre-trained model outperforms other approaches and score the highest in all sub-tasks. We rank on the leader board at the evaluation phase are 26, 26, 25, and 9 through 0.9564 F-score, 0.5709 RMSE, 0.4888 F-score, and 0.4467 RMSE results, respectively, for each of the first, second, third, and fourth sub-task, respectively.

## 1 Introduction

Dealing with natural languages has long been a challenge and an interesting topic for researchers (Chowdhury, 2003). Understanding and generating languages is part of natural language processing (NLP) (Nadkarni et al., 2011). Recently, the language model is able to deal with sequence-to-sequence problems such as question and answer, translation, multiple choice. In addition, it is able to capture complex relationships, semantic meaning, word meaning disambiguation, and word aspect-based (Deng and Liu, 2018). Humorous text is one of the important things we watched every day. It is commonly used to express an opinion on issues (societal, political, sports, and economic), whether in posts on social media platforms, or as advertising for a specific product (Kramer, 2011). In addition, the humor in the text makes the text complex in terms of interpretation and understanding of the text. Because of the manipulation of the meaning of words and the way the text is written to express the sense of humor in the words. On the other hand, understanding the humorous in the text varies according to the age or gender of the person, or even according to the culture, social status and mentality of the person(Goel and Dolan, 2007). In this task, a dataset was collected in the English language that represents humor and joke in the text and words. We participated in this task to build an approach capable of distinguishing a text that is humorous or not. Here we have explicitly used pre-trained models that deal with the concept of contextual text such as Bert (Devlin et al., 2018), Roberta (Liu et al., 2019), and XLM-Roberta (Conneau et al., 2019). In addition, as a baseline we worked on training the dataset by the Bert embedding layer and extracting weights to feed it into the Bi-LSTM and Dense layers.

In all sub-tasks we used as a baseline BiLSTM (Graves and Schmidhuber, 2005) layer with a BERT embedding layer, as well as, pre-trained models such as Bert (Devlin et al., 2018), Roberta (Liu et al., 2019), and XLM-Roberta (Conneau et al., 2019).

In all sub-tasks, Roberta model showed superiority compared to other approaches. We ordered according to the official results among 36 participating teams. In the first sub-task, we achieved 26th rank with an 0.9564 F-score result. On the second sub-task, ranked 26th with a score of 0.5709 RMSE. A third sub-task placed 25th with 0.4888 F-score (Sokolova et al., 2006). The last sub-mission we took the 9th rank with a score of 0.4467 RMSR (Chai and Draxler, 2014). The remainder of this paper is organized as follows: Background in Section 2. The properties of the dataset and the system in section 3. Section 4 explains the experiment and analyzing results. The last section 5 shows

*Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 1102–1107
Bangkok, Thailand (online), August 5–6, 2021. ©2021 Association for Computational Linguistics

conclusions and future work.

## 2 Background

In (Hossain et al., 2019) developed a new humor corpus, which consist of 15,095 news headlines in English. They substituted the headlines with few words to be funny. Also, (Li et al., 2020) used attention-based with bi-directional long short-term memory (AttBiLSTM) to classify slang language into negative humor or positive humor. In (Annamoradnejad, 2020) utilized a BERT embedding layer with several parallel hidden layer to categorize 200K humorous sentences whether (positive or negative). While (Fan et al., 2020) used two kinds of attention mechanisms (internal and external) to capture sense of humor in words. Most of the previous works came to predict the humor polarity (positive, negative) or the humor rating (range values) in the text. However, this research addressing the humor and offensive score detection.

## 3 Methodology

### 3.1 Task Description

We worked with four sub-tasks provided by SemEval-2021 [1], in task-1 divided into (a, b and c sub-tasks). Each sub-task related to the other. Moreover in task-2 has one sub-task (a). In general, Sub-task-1a will predict whether the text expresses a humorous or not (binary classification problem 1, 0). Sub-task-1b if the text is considered a humorous, will predict how humorous it is from 0 to 5 values (regression problem). Sub-task-1c If the text is a humorous, we would predict if it is controversial or not (binary classification problem 1, 0). Sub-task-2a will predict the offense.

### 3.2 Data-set

A dataset consists of a set of texts and each text has four categories (is-humor, humor-rating, humor-controversy, offense-rating) in English language (Meaney et al., 2021). Each text asked by 20 annotators to label each category of the text. As well as, the annotators come from different gender and age groups. For is-humor and humor-controversy categories were taken the majority of the classes by 20 annotators as label for each text. Whereas, humor-rating and offense-rating categories take the average of rating classes between range 1 and 5 over 20 annotators as a label for each text.Table 2 shows examples of the training dataset per text with the four classifications for each category. Moreover, in humor-rating and humor-controversy, we noted the categories have many NaN values, because if is-humor the majority of the classes were not classified as humor which means 0 label, so the remaining categories of humor are NaN values. Therefore, we need to remove the NaN values from each category as pre-processing the dataset before training the models. Table 1 shows the total number in the training, development and testing dataset for each category.

| **Dataset** | **Is-H** | **H-R** | **H-C** | **O-R** |
|---|---|---|---|---|
| Training | 8000 | 4932 | 4932 | 8000 |
| Development | 1000 | 632 | 632 | 1000 |
| Testing | 1000 | 615 | 615 | 1000 |

Table 1: The total number for each category (is-humor, humor-rating, humor-controversy, offense-rating) after removing NaN values.

### 3.3 System overview

The proposed system focused on pre-trained transformer models, we Moreover applied some techniques that represent embedding words and feeding them into long short-term memory (LSTM) layers to train the data-set. Through all of the sub-tasks, the highest score was via the Roberta model. It is one of the powerful models pre-trained on a huge data-set and complex architecture. As well, it was released by Facebook and designed base on the BERT model that was released by Google. All pre-trained models are capable of handling long text dependencies and capturing features and relationships. Furthermore, the structure of pre-trained models that involve encoder-decoder (Cho et al., 2014) is enabled to deal with sequence-to-sequence (Sutskever et al., 2014) tasks. In addition, BERT-Large (Devlin et al., 2018) and Roberta-Large (Liu et al., 2019) models consisted of 24 layers, 1024 hidden units of output word embedding, and 16 head attention layers, where both models have the same layered structure but differ in the method of approach to training and the volume of data used to train each model.

There are two approaches used to train a BERT model 1- Masked Language Model (MLM) is masking some tokens of the training dataset with a [mask] symbol and try to predict the token. 2 - Next Sentence Prediction (NSP) is training the dataset by assigning 1's for neighboring sentences

[1] https://competitions.codalab.org/competitions/27446#learn_the_details

| Text | Is-H | H-R | H-C | O-R |
|---|---|---|---|---|
| TENNESSEE: We're the best state. Nobody even comes close. *Elevennessee walks into the room* TENNESSEE: Oh shit | 1 | 2.42 | 1 | 0.2 |
| I got REALLY angry today and it wasn't about nothing, but you're going to have to take my word for that. | 0 | Nan | Nan | 0.15 |
| Told my mom I hit 1200 Twitter followers. She pointed out how my brother owns a house and I'm wanted by several collection agencies. Oh ma! | 1 | 2.11 | 1 | 0 |

Table 2: An example illustrating the features of a training dataset whether a humor or offense. If Is-humor class is 0 then the Humor-rating and Humor-controversy classes are Nan values.

and 0's for randomly chosen sentences. In contrast, Roberta used the MLM model approach for the training phase, as well as trained on a huge dataset compared to the BERT model.

Moreover, we tried the XLM-Roberta-Large pretrained model (Conneau et al., 2019), which has 550M parameters with 24-layers of architecture. In addition, it consists of 1024 of the output hidden-state embedding, 4096 of feed-forward hidden-state, and 16 of head attentions. The model has Trained on 2.5 TB of newly created clean CommonCrawl data that supports 100 languages.

On the other hand, this research exploits BERT embedding to represent text. Where the weights were extracted by training the dataset on the BERT embedding layer and then feeding them into a BI-LSTM layer of 128 units (Graves and Schmidhuber, 2005). Moreover, We used the dropout layer with 0.3 ratios, the max-pool layer, then passing the information into a dense layer with 64 units. In the last layer for classification tasks, the final dense layer is 2 hidden output units with a sigmoid activation function, and for regression, one unit output in the final dense layer. The Figure 1 shows the model architecture used for prediction label on classification and regression tasks.

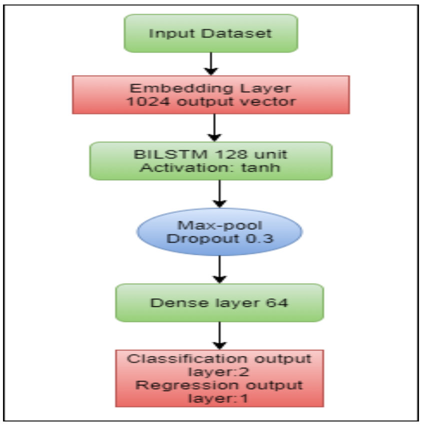
Input Dataset
Embedding Layer 1024 output vector
BILSTM 128 unit Activation: tanh
Max-pool Dropout 0.3
Dense layer 64
Classification output layer:2 Regression output layer:1

Figure 1: An illustration of the proposed model architecture.

## 4 Experimental and Results

In the experimental phase, the dataset was divided into three parts (training, development, and testing). We used the training dataset to train the model, and the development dataset to fine-tune the model to capture the best hyper-parameters without occurring over-fitting or under-fitting the model. Moreover, we used the test data set to check the performance of the model with an unseen dataset and to ensure the generalizability of the model. However, to perform the experiments we used collaborative google Colab as a platform, which provides a number of GPUs available for use with modest memory size [2]. In addition, in our experiments with pretrained models, we used the transformers library that is based on the PyTorch language and allows you to fine-tune the models and train them on your own dataset [3]. In training the model, we did not use any pre-processing technique in the entered dataset. Although, there are some symbols, upper and lower case letters, misspellings, and some abbreviations in the dataset. However, We did not treat these issues in dataset, where the dataset is trained as it is. In order, for the model to be more realistic and robust in dealing with the real dataset. As well as, the model might deal with those cases as features for each case in the dataset for the model learning phase. Just in pre-processing phase, we needed to remove NaN values in both sub-task (1B and 1C ). In order to test the performance of the approaches used in this task, where each sub-task has a metric that meets the type of output of each sub-task such as regression metric or classification metric. Accuracy and F-score metrics were a measure of the

[2] https://colab.research.google.com
[3] https://huggingface.co/transformers/

performance in sub-task-1a and sub-task-1c. Likewise, the RMSE metric was a measure of outcome performance in both sub-task-1b and sub-task-2c. In the process of model tuning, we tried several hyper-parameters, where the batch size was fixed 8, and the Adam optimizer function was used on all experiments. Furthermore, we applied several learning-rates in the range 1e-5, 4e-5, 1e-6, 3e-6 and a different number of epoch 2, 4, 8, 12 epochs. The table 3 shows the main experiments among many models with different LRs and Epochs for each sub-task.

| Sub-task | Model | Epoch | LR |
|---|---|---|---|
| 1-A | Roberta-Large | 4 | 1e-6 |
| | Roberta-Large | 4 | 3e-6 |
| | Roberta-Large | 8 | 1e-6 |
| | XLM-Roberta-Large | 8 | 1e-6 |
| | BERT-Large-Cased | 8 | 1e-6 |
| | BERT embedding + BiLSTM | ES | 2e-5 |
| 1-B | Roberta-Large | 8 | 1e-6 |
| | Roberta-Large | 8 | 1e-5 |
| | Roberta-Large | 12 | 1e-5 |
| | Roberta-Large | 2 | 3e-5 |
| | BERT-Large-Cased | 8 | 3e-5 |
| | BERT embedding + BiLSTM | ES | 2e-5 |
| 1-C | Roberta-Large | 8 | 1e-6 |
| | Roberta-Large | 8 | 1e-5 |
| | Roberta-Large | 8 | 8e-5 |
| | XLM-Roberta-Large | 8 | 8e-5 |
| | BERT-Large-Cased | 8 | 8e-5 |
| | BERT embedding + BiLSTM | ES | 2e-5 |
| 2-A | Roberta-Large | 8 | 1e-5 |
| | Roberta-Large | 8 | 3e-5 |
| | Roberta-Large | 4 | 1e-5 |
| | Roberta-Large | 12 | 1e-5 |
| | BERT-Large-Cased | 8 | 1e-5 |
| | BERT embedding + BiLSTM | ES | 2e-5 |

Table 3: The models applied and hyper-parameters used. (ES denotes to early-stopping technique)

### 4.1 Result

Roberta achieved high-performance results compared to other approaches, that exhibit his ability to capture traits and distinguish between labels. The table 4 presents the best results for both development and evaluation level results, as well as the best hyper-parameters selected based on the experimental phase for each sub-task. In sub-task-1A Roberta-Large achieved high scores in a binary classification problem compared to the other models, where we scored 26 at F-score metrics in the evaluation phase for our ranking on the leader-board. While in the sub-task-2B also Roberta achieved acceptable results in the regression problem and outperformed the other models, as we ranked on the Leader Board 26 at RMSR metric. For the rest of the other sub-tasks, sub-task -3C is treated as a binary classification, which we achieved 25 rank in evaluation phase by F-score. In the last sub-task, our rank was 9 for an RMSR metric at the evaluation phase on the leader board.

#### 4.1.1 Error Analysis

This section presents some analyzes to clarify the outcomes and limitations model of each sub-task. Figure 2 represents the confusion matrix for each label is given in sub-task-1A which is a classification problem. The figure shows the number of cases which the actual label matches the predicted label (y = ŷ) which is 946 in total. While the number of labels that differ (y != ŷ) that the model could not predict the label, it is 54. In the square that represents 31 false positives, we can see that it is a little more than the square that represents 23 false positives. This is because the training dataset is slightly biased towards label 1, which is 4,932 out of 8,000, while label 0 makes up 3,068 of the training data set.

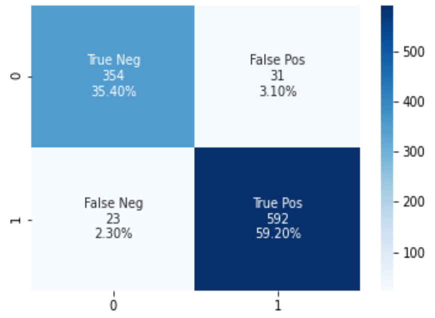

Figure 2: An illustration of the confusion matrix for sub-task-1A.

Moreover, the sub-task-1B represented in figure 3. We applied the round function to obtain integer numbers and categories of labels to display, which shows four values existing in this task as continuous labels in range 0 - 4. In the figure, the label shows a label 2 obtained the highest match in the model between the actual labels and the predicted

| Sub-task | Model | Epoch | LR | Development result | Evaluation result |
|---|---|---|---|---|---|
| 1-A | Roberta-L | 8 | 1e-6 | 0.9426-F1 & 0.9270-Acc. | 0.9564-F1 & 0.9460-Acc. |
| 1-B | Roberta-L | 2 | 3e-5 | 0.518-RMSE | 0.5709-RMSE |
| 1-C | Roberta-L | 8 | 8e-5 | 0.5493-F1 & 0.5585-Acc. | 0.4888-F1 & 0.5545-Acc. |
| 2-A | Roberta-L | 12 | 1e-5 | 0.5209-RMSE | 0.4467-RMSE |

Table 4: The best results gained for both development and evaluation level with hyper-parameters chosen.

labels, while the labels 0, 1, 4, the model could not recognize them in the prediction phase. This is due to the size of the training dataset is a little for each label compared to 2, 3 labels. The size of the dataset in 0, 1, and 4 labels in the training dataset are 16, 410, and 47 respectively. On the other hand, label 2 is repeated 2835 times and 3 is repeated 1624 times in the training dataset.

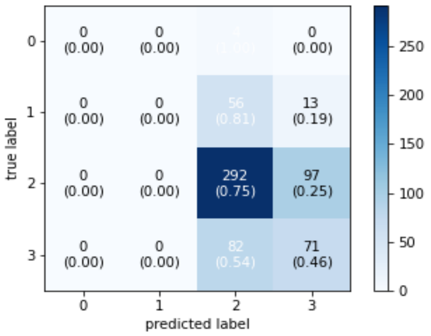

Figure 3: An illustration of the discretization confusion matrix for sub-task-1B.

A third sub-task, which is a binary classification problem. The figure 4 shows the model is able to recognize label 0 a little more than label 1, but in general, the model is not learned well (high biased). The number of cases for both (0 and 1) labels in the training dataset were 2467 and 2465 almost equal, respectively.

Finally, in the last sub-task-2C, we needed to use a round function to approximate continuous values to discrete values. However, the diameter of the figure 5 clearly shows the highest label to the lowest label distinguished by the model. Where the values are logically acceptable compared to the number of cases for each label in the training dataset, which are 5737, 1043, 623, 364, 214, and 19 frequency for each of 0, 1, 2, 3, 4, 5 labels respectively.

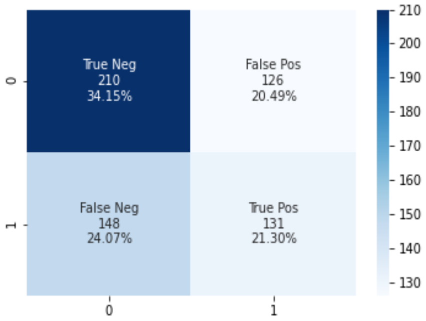

Figure 4: An illustration of the confusion matrix for sub-task-1C.

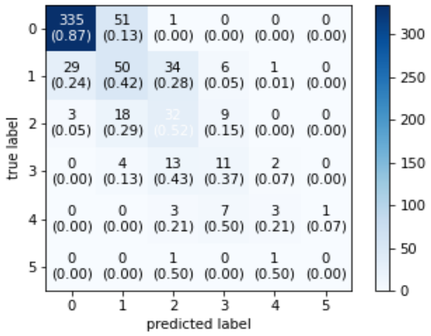

Figure 5: An illustration of the discretization confusion matrix for sub-task-2A.

## 5 Conclusion

In this paper, we presented several approaches that addressed four sub-tasks. We obtained high scores using a pre-trained Roberta model for each sub-task. In the first sub-task, predicting if the text is humorous or not, we gained a 0.9564 F-score. While in the second sub-task, finding a humorous text representation rate from 0 to 5, that was got a 0.5709 RMSE. A third sub-task, verification of the text is controversial or not, obtained a 0.4888 F-score. The last sub-task is to find the offensive rate in the text for the range of 0 to 5, which achieved 0.4467 RMSE. For future works, we are going to do more experiments and using ensemble technique to enhance the results. Moreover, adding more dataset with the original to treat the biased label.

## References

Issa Annamoradnejad. 2020. Colbert: Using bert sentence embedding for humor detection. *arXiv preprint arXiv:2004.12765*.

Tianfeng Chai and Roland R Draxler. 2014. Root mean square error (rmse) or mean absolute error (mae)?–arguments against avoiding rmse in the literature. *Geoscientific model development*, 7(3):1247–1250.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Gobinda G Chowdhury. 2003. Natural language processing. *Annual review of information science and technology*, 37(1):51–89.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

Li Deng and Yang Liu. 2018. *Deep learning in natural language processing*. Springer.

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

Xiaochao Fan, Hongfei Lin, Liang Yang, Yufeng Diao, Chen Shen, Yonghe Chu, and Yanbo Zou. 2020. Humor detection via an internal and external neural network. *Neurocomputing*, 394:105–111.

Vinod Goel and Raymond J Dolan. 2007. Social regulation of affective experience of humor. *Journal of cognitive neuroscience*, 19(9):1574–1580.

Alex Graves and Jürgen Schmidhuber. 2005. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural networks*, 18(5-6):602–610.

Nabil Hossain, John Krumm, and Michael Gamon. 2019. " president vows to cut¡ taxes¿ hair": Dataset and analysis of creative text editing for humorous headlines. *arXiv preprint arXiv:1906.00274*.

Elise Kramer. 2011. The playful is political: The metapragmatics of internet rape-joke arguments. *Language in Society*, pages 137–168.

Da Li, Rafal Rzepka, Michal Ptaszynski, and Kenji Araki. 2020. Hemos: A novel deep learning-based fine-grained humor detecting method for sentiment analysis of social media. *Information Processing & Management*, 57(6):102290.

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

J.A. Meaney, Steven R. Wilson, Luis Chiruzzo, Adam Lopez, and Walid Magdy. 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*.

Prakash M Nadkarni, Lucila Ohno-Machado, and Wendy W Chapman. 2011. Natural language processing: an introduction. *Journal of the American Medical Informatics Association*, 18(5):544–551.

Marina Sokolova, Nathalie Japkowicz, and Stan Szpakowicz. 2006. Beyond accuracy, f-score and roc: a family of discriminant measures for performance evaluation. In *Australasian joint conference on artificial intelligence*, pages 1015–1021. Springer.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.