

Leveraging Ensemble Voting and Fine-Tuning Strategies in Pre-Trained Transformers to Detect Prejudicial Tweets and Hurtful Humour

Jay Kaoshik^{1,*}, Sharmila Banu Kather¹

¹*School of Computer Science and Engineering,
Vellore Institute of Technology, Vellore, 632014, Tamil Nadu, India*

Abstract

The contents of this paper describe our system which was submitted to the HURtful HUmour Detection (HUHU) task at IberLEF 2023. The proposed system achieved a F1 score of **0.781** for Task 1 (binary classification) using an ensemble of pre-trained transformers which included DistilBERT Cased, XLM-RoBERTa Spanish, RoBERTuito Cased, mBERT Cased and BERT Cased. A Macro F1 score of **0.739** was obtained for Task 2A (multi-label classification) using ensemble voting from mBERT Cased, BETO Cased, BETO Uncased, DistilBERT Spanish and RoBERTa. For the final Task 2B (regression), our proposed system achieved a RMSE score of **0.938** when the pre-trained transformers were fine-tuned with a fully connected layer (regression head) over the pre-final layer. For the regression task, an ensemble of BETO Cased, BETO Uncased, ALBERT Spanish, DistilBERT Spanish and RoBERTuito Uncased was used. Our proposed ensemble system ranked in the **Top 10** systems team-wise for Subtasks 1 and 2A without using knowledge from a high performing baseline i.e. Bloom-1b1 and without any form of data pre-processing or augmentation.

Keywords

Transformers, Ensemble Learning, Hurtful Humour Detection, Natural Language Processing

1. Introduction

HURtful HUmour Detection 2023 [1] is a task that seeks to determine the existence of any negative pre-judgement or stereotypes against significant social groups and detect hurtful humour directed towards such communities. In this process, we attempt to analyze the tweets and their composition to understand which segments of the corpus contribute to humour. This task was further classified to three sub-tasks: Binary classification to detect if the tweets are humorous or not, multi-label classification to determine hurtful humour targeted at specific communities and groups and the final regression task to determine the extent of prejudice on a scale of 1-5. Transformers have been a significant breakthrough in the field of Natural Language Processing by achieving state-of-the-art results in various tasks such as language translation, text generation, sentiment analysis, and more. Transformers have transformed sentiment analysis by offering contextual word representations, swiftly managing sequential data, successfully capturing contextual dependencies, facilitating transfer learning, and displaying flexibility across multiple sentiment analysis tasks. These developments have produced sentiment analysis models that are more precise and strong and are better able to interpret sentiment in natural language text. Ensemble voting on the other hand helps mitigate the risk of relying on a single model's biases as it combines the predictions by either a majority voting (for classification tasks) or averaging (for regression tasks).

IberLEF 2023, September 2023, Jaén, Spain


*Corresponding author.

✉ jaynيتين.kaoshik2020@vitstudent.ac.in (J. Kaoshik); sharmilabanu.k@vit.ac.in (S. B. Kather)

🌐 <https://research.vit.ac.in/researcher/sharmila-banu-k> (S. B. Kather)



© 2023 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

2. Related Work

2.1. Recent Advancements

Contemporary research in the fields of sentiment analysis and ensemble voting have widely contributed to Hate Speech Detection [2]. The use of cross-lingual embeddings fine tunes neural networks and some studies [3] leverage a hybrid voting-boosting ensemble learning method which bring down computing time and uses sequential structure to optimize utilization of erroneous data by employing cross segmentation approach. Leveraging their self-attention mechanisms, transformers proficiently capture and discern sentiment-bearing words or phrases in sentences while considering the holistic context. Consequently, sentiment classification accuracy has markedly improved, encompassing the identification of positive, negative, or neutral sentiment. Transformers excel at capturing intricate contextual dependencies. Ensemble voting, on the other hand, elevates overall performance of a task by combining predictions from multiple models. Current research on existing ensemble voting methods [4] shows that weighted majority voting performs better compared to the traditional simple majority voting on the Twitter Sentiment Analysis Dataset and Stanford Twitter Sentiment Corpus among others. Pre-trained transformers such as the DistilBERT have been proved to outperform some baseline attention based recurrent neural networks for tasks of hate speech detection [5]. The proposed model in [6] can readdress the softmax probabilities of the participating classifiers depending on their primary outcomes. This weighting technique has enabled the model to outdo the simple average ensemble.

2.2. Synergistic Sentence Classification: Transformers and Ensemble Voting

Transformers are composed of layers of self-attention mechanisms and an encoder-decoder mechanism. The input corpus is converted to feature embeddings which capture semantic and contextual information. A self attention mechanism allows the system to analyze the impact of different tokens by recording long range dependencies. The transformer contains self-attention sub-layers and feed-forward neural network sub-layers which are responsible for parallel computation of the input sequence. In some tasks, such as language translation, transformers employ a decoder component. The decoder predicts the target sequence based on the encoded representation generated by the encoder. By giving consideration to other words in the input sentence, self-attention enables the transformer model to identify word dependencies. It stimulates attention scores that account for the significant connections between each word and its neighbours. The word representations are then updated using these attention scores.

Ensemble voting uses the concepts of weighted majority voting and soft voting [7]. Let \hat{y} denote the weighted majority vote. It can be computed by assigning a weight w_j to a classifier C_j . A denotes the set of distinctive class labels and X_A is the characteristic discriminant.

$$\hat{y} = \underset{i}{\operatorname{argmax}} \sum_{j=1}^m \{w_j X_A\} \{C_j(x) = i\}$$

In our tasks, we use hard voting & averaging. The classes which receive the majority of the votes from all the constituent models of an ensemble system are taken into account into the final prediction. The concept of soft voting can also be considered for analyzing the target variables. It contemplates the probability estimates or confidence scores provided by each model and the probabilities for each class label from all models are averaged. The class label with the highest averaged probability is selected as the final prediction. Soft voting is particularly beneficial when the models in the ensemble are well-calibrated and provide reliable probability.

3. Experimental Setup

3.1. Dataset Description

The first batch of data provided by the organizers was the Training Set. This set incorporated 2671 spanish tweets, the 6 labels associated with the tweet and additionally an index label to uniquely identify the tweet. The data used to evaluate the fine-tuned models was a subset of this Training Set. The ratio of split used for Training and Evaluation sets was 80:20. The split was stratified in case of Task 1 (HUrTful HUmour Detection) to ensure that there were equal proportion of positive and negative samples in the training and evaluation sets.

The actual Test Set which was used to determine the final ranks of proposed systems for all tasks had 778 tweets and the indexes associated with those tweets. The columns present in the corpus utilized for training and testing have been described in detail below.

- `index` : a 17-bit integer acting as a unique identification to the actual tweet text.
- `tweet` : the contents of the spanish tweet as plain text for which prediction is needed.
- `humor` : a binary valued label indicating if a prejudicial tweet is intended to cause humour.
- `prejudice_woman` : a binary label indicating if the tweet targets women and feminists.
- `prejudice_lgbtiq` : a binary label indicating if the tweet targets LGBTIQ community.
- `prejudice_inmigrant_race` : a binary valued label indicating if the tweet targets immigrants and racially discriminated people.
- `gordofobia` : a binary valued label indicating if the tweet targets overweight people.
- `mean_prejudice` : a real value between 1 and 5 depicting the average prejudice value.

The Training and Testing Data can be obtained from [Zenodo \(Record 7967255\)](#) [8].

3.2. Task Description

The HuHu challenge [1] proposes three sub-tasks which have been briefly summarized below:

- **Subtask 1 - HUrTful HUmour Detection**

The first subtask aims to determine whether a prejudicial tweet is intended to cause humour. The task in hand is to distinguish between tweets that use humour to express prejudice and the tweets that express prejudice without using humour. For this, the systems will be evaluated and ranked employing the F1-measure over the positive class.

- **Subtask 2A - Prejudice Target Detection**

The second subtask takes into account the minority groups analyzed, i.e, Women and feminists, LGBTIQ community, Immigrants and racially discriminated people, and overweight people. The task aims to identify the targeted groups on each tweet as a multilabel classification problem. The metric employed for evaluation here is Macro-F1.

- **Subtask 2B - Degree of Prejudice Prediction**

The third subtask aims to predict an average rating for each tweet which indicates the degree of prejudice. This regression task asks the systems to rate the tweet on a continuous scale (from 1 to 5) to evaluate how prejudicial the message is on average among minority groups. The evaluation metric employed here is the Root Mean Squared Error.

3.3. Training Environment

All models in our ensemble were fine-tuned using the AdamW optimizer with a learning rate of $4e-5$ and a batch size of 8. These models were trained on the NVIDIA Tesla V100 GPU.

4. Subtask 1 - HUrTful HUmour Detection

The scores obtained on the **Evaluation Set** (random stratified split of training set) for this binary classification task have been mentioned in Table 1 and Table 2. The baselines provided by the organizers for this task include the BLOOM-1b1 transformer, the BETO transformer [9] and SVM + Character Level n-gram which achieved F1 Scores of 0.789, 0.759 and 0.679 respectively on the Test Set. The general idea for submission was to encompass ensemble voting from various fine-tuned pre-trained transformers instead of just using a single high performing transformer like BETO [9] which outperformed the ensembles in Task 2A and 2B by a considerable margin. The aggregated representation of [CLS] token is passed to a dense layer over which we apply a sigmoid activation function; thus converting logits into normalized probabilities for classification.

Table 1
Individual Performance of Pre-Trained Transformers with Fine-Tuning on Evaluation Set

Reference	Transformer	F1 Score
[10]	BERT Cased	0.76785
[10]	BERT Uncased	0.74074
[10]	BERT Spanish Hate Speech	0.75842
[10]	mBERT Cased	0.77511
[10]	mBERT Uncased	0.74772
[9]	BETO Cased	0.76191
[9]	BETO Uncased	0.73456
[9], [11]	BETO Sentiment Analysis	0.72928
[12]	AlBERT v2	0.69594
[13]	AlBERT Spanish	0.71929
[13]	AlBERT Tiny Spanish	0.71811
[14]	DistilBERT Cased	0.77936
[14]	DistilBERT Uncased	0.71826
[13]	DistilBERT Spanish	0.72141
[15]	RoBERTa	0.75281
[16]	XLM-RoBERTa Spanish	0.77945
[11]	RoBERTa Sentiment Analysis	0.75362
[11], [17], [18]	RoBERTuito Sentiment Analysis	0.75776
[17]	RoBERTuito Uncased	0.77809
[17]	RoBERTuito Cased	0.77808
[19]	DeBERTa	0.76571
[20]	XLNet Cased	0.74033
[21]	ELECTRA Small	0.69938
[21]	ELECTRA Discriminator	0.74643
[22]	CamemBERT	0.74137
[14]	mDistilBERT Cased	0.75942

The proposed ensemble system is based on majority voting from 5 pre-trained transformers which have been listed in Table 2. We fine-tuned each of these models for 3 epochs and the whole training process took about 12-13 minutes on the V100. The ensemble achieved a F1 Score of 0.8 on the Evaluation Set. The voting mechanism used for this task has been briefly described below.

$$k = \sum_{i=1}^m \frac{\{P_i\}}{m}, \quad \hat{y} = \begin{cases} 0 & \text{if } 0 \leq k < 0.5 \\ 1 & \text{if } 0.5 < k \leq 1 \end{cases}$$

k - real valued number, P_i - predicted class label from model i , m - number of models ($m\%2=1$ always), \hat{y} - predicted ensemble voting output

Table 2

Performance of Proposed Ensemble with Fine-Tuning and Majority Voting on Evaluation Set

Ensemble System	F1 Score
DistilBERT Cased + XLM-RoBERTa Spanish + RoBERTuito Cased + mBERT Cased + BERT Cased	0.8

5. Subtask 2A - Prejudice Target Detection

The scores obtained on the **Evaluation Set** (random split of training set) for this multi-label classification task have been mentioned in Table 3 and Table 4. The baselines provided by the organizers for this task include the BETO transformer [9] and a SVM + Character Level n-gram which achieved Macro F1 Scores of 0.760 and 0.603 respectively on the Test Set. BETO Cased [9] outperformed all individual models with a Macro F1 Score of 0.93504 on the Evaluation Set as seen from Table 3. The ensemble system proposed in Table 4 achieved a Macro F1 Score of 0.94351 on the Evaluation Set. On the Test Set, our ensemble system had a F1 Score of 0.739 as opposed to the 0.760 Macro F1 score by BETO baseline. This also indicates that a system using ensemble voting from high performing pre-trained transformers might not always rank above the individual best transformer(s) in composition of the ensemble (BETO [9] in our case).

Table 3

Individual Performance of Pre-Trained Transformers with Fine-Tuning

Reference	Transformer	Macro F1 Score
[10]	BERT Cased	0.87749
[10]	BERT Uncased	0.86443
[10]	mBERT Cased	0.90448
[10]	mBERT Uncased	0.88525
[9]	BETO Cased	0.93504
[9]	BETO Uncased	0.91617
[12]	ALBERT v2	0.80617
[13]	ALBERT Spanish	0.87299
[13]	ALBERT Tiny Spanish	0.84810
[14]	DistilBERT Cased	0.87969
[14]	DistilBERT Uncased	0.86065
[13]	DistilBERT Spanish	0.90543
[15]	RoBERTa	0.89597
[20]	XLNet Cased	0.88661
[21]	Electra Small	0.82617
[21]	Electra Discriminator	0.86549
[22]	CamemBERT	0.82358
[14]	mDistilBERT Cased	0.88797

The proposed ensemble system is based on majority voting from 5 pre-trained transformers which have been listed in Table 4. We fine-tuned each of these models for 4 epochs and the whole training process took about 11-12 minutes on the V100. The voting mechanism used for this multi-label binary classification task has been briefly described below.

$$(K_j)_{j=1}^n = \left(\sum_{i=1}^m \frac{P_{i,j}}{m} \right)_{j=1}^n, \quad (\hat{K}_j)_{j=1}^n = \begin{cases} 0 & \text{if } 0 \leq K_j < 0.5 \\ 1 & \text{if } 0.5 < K_j \leq 1 \end{cases}$$

K_j - Real valued number denoting intermediary output for target label j computed across m

systems, $\mathbf{P}_{i,j}$ - predicted class label from model i for category j , \mathbf{m} - number of models ($m\%2=1$ always), n - number of labels to be predicted, $\hat{\mathbf{K}}_j$ - predicted voting output for target j

Table 4
Performance of Proposed Ensemble with Fine-Tuning and Majority Voting on Evaluation Set

Ensemble System	Macro F1 Score
mBERT Cased + BETO Cased + BETO Uncased + DistilBERT Spanish + RoBERTa	0.94351

6. Subtask 2B - Mean Prejudice Detection

For the regression task of predicting a mean prejudice score for each tweet, we use a regression head i.e. a fully connected linear layer. This regression head uses pooled representations of the [CLS] token extracted from the hidden state sequence produced by various transformers. The scores obtained on the **Evaluation Set** (random split of training set) for this regression task have been mentioned in Table 5 and Table 6. The baselines provided by the organizers for this task include the BETO transformer [9], a SVM + Character Level n-gram and the BLOOM-1b1 transformer which achieved RMSE scores of 0.874, 0.907 and 0.915 respectively on the Test Set.

Table 5
Individual Performance of Pre-Trained Transformers with Fine-Tuning on Evaluation Set

Reference	Transformer	RMSE
[10]	BERT Cased	0.78910
[10]	BERT Uncased	0.79969
[10]	mBERT Cased	0.73108
[10]	mBERT Uncased	0.78555
[9]	BETO Cased	0.69568
[9]	BETO Uncased	0.71792
[12]	AlBERT v2	0.73672
[13]	AlBERT Spanish	0.72136
[13]	AlBERT Tiny Spanish	0.78347
[14]	DistilBERT Cased	0.78780
[14]	DistilBERT Uncased	0.80084
[13]	DistilBERT Spanish	0.71911
[15]	RoBERTa	0.82665
[17]	RoBERTuito Uncased	0.71735
[20]	XLNet Cased	0.74883
[21]	Electra Small	0.77367
[21]	Electra Discriminator	0.85276
[22]	CamemBERT	0.77892
[14]	mDistilBERT Cased	0.74169

The final submission was performed by the first ensemble combination in Table 6 i.e. the one with the lowest RMSE on Evaluation Set. We fine-tuned each of these models for 5 epochs and the whole training process took about 10-11 minutes on the V100 for a 5-ensemble and 6-7 minutes for a 3-ensemble system. The linear average voting schema has been briefed below.

$$\hat{y} = \sum_{i=1}^m \frac{y_i}{m}$$

y_i - Real valued number denoting the predicted output from model i for the regression task, m - number of models ($m \geq 1$ always), \hat{y} - mean score from ensemble of m models

Table 6
Performance of Proposed Ensemble with Fine-Tuning and Averaging on Evaluation Set

Ensemble System	RMSE
BETO Cased + BETO Uncased + AIBERT Spanish + DistilBERT Spanish + RoBERTuito Uncased	0.65687
BETO Cased + mBERT Cased + AIBERT Spanish + DistilBERT Spanish + RoBERTuito Uncased	0.66865
BETO Cased + BETO Uncased + RoBERTuito Uncased	0.67006

7. Results and Conclusion

From Test Set Results of Subtask 2B, we infer that BETO had a RMSE of 0.874 as opposed to our best ensemble which had a RMSE of 0.938 despite BETO being given superior weightage in the ensemble due to its dual presence from cased and uncased versions. The same observation is reflected by results of Subtask 2A as well. This shows that Ensemble Voting might not always benefit tasks like Regression and Multi-Label Classification but can result in a significant performance boost for a rather simplified task of Binary Classification as seen from results of Subtask 1.

The run submitted for ranking was meant to demonstrate the performance of ensemble voting rather than to obtain a better rank from a single high performing baseline like BETO which was either way experimented with individually in all the subtasks. We present a comparative summary of results on the Test Set in Table 7, Table 8 and Table 9 for Subtasks 1, 2A and 2B respectively.

Table 7
Subtask 1: Performance of Proposed Ensemble with Fine-Tuning and Majority Voting on Test Set

Submitted System	F1 Score
First Place Solution	0.820
Second Place Solution	0.799
Third Place Solution	0.796
BLOOM-1b1	0.789
DistilBERT Cased + XLM-RoBERTa Spanish + RoBERTuito Cased + mBERT Cased + BERT Cased	0.781
BETO	0.759
SVM-3gram-char	0.679
AllTrue	0.492

Table 8
Subtask 2A: Performance of Proposed Ensemble with Fine-Tuning and Majority Voting on Test Set

Submitted System	Macro F1 Score
First Place Solution	0.796
Second Place Solution	0.783
Third Place Solution	0.778
BETO	0.760
mBERT Cased + BETO Cased + BETO Uncased + DistilBERT Spanish + RoBERTa	0.739
SVM-3gram-char	0.603
AllTrue	0.482

Table 9

Subtask 2B: Performance of Proposed Ensemble with Fine-Tuning and Averaging on Test Set

Submitted System	RMSE
First Place Solution	0.855
Second Place Solution	0.874
BETO	0.874
SVR-3gram-char	0.907
BLOOM-1b1	0.915
BETO Cased + BETO Uncased + ALBERT Spanish + DistilBERT Spanish + RoBERTuito Uncased	0.938

In all the Subtasks, we tried to gain a performance boost either in F1 Score or RMSE using ensemble voting from pre-trained transformers. For Task 1, i.e. the main task of Hurtful Humour Detection, we outperformed most of the systems proposed without using the knowledge from a significant baseline i.e. the Bloom-1b1 Transformer. We also had a decent margin between other baselines like BETO and SVM-3gram-char. For Subtask 2A as well, our ensemble performed decently well when compared to BETO and other proposed systems. In case of Subtask 2B, we see a sharp decrease in performance when seen relative to expectations from Subtask 1 and Subtask 2A. We also experimented with BETO for all subtasks which ranked highly in 2A and 2B subtasks. All the models were trained with raw text and no form of pre-processing /augmentation was used.

8. Scope for Further Research

The proposed future work includes a comprehensive analysis of other voting methodologies that can be implemented. Weighted majority voting could be used to assign weights to a classifier based upon its confidence and reliability. Each classifier would generate a distinctive set of label predictions and the final prediction would be determined by analysing the predictions of various classifiers aggregated across their respective weights. Another voting method that can be implemented is the soft voting method which takes into account the probability scores of various systems. Each classifier predicts the probability of various labels which are computed in a way that the highest probability is assigned as the target prediction probability. Further scope of research could include the tasks of data preprocessing such as handling negations, removal of emojis, emoticons, slangs, abbreviations and stop words, handling imbalanced classes and performing stemming and lemmatization.

Acknowledgments

The authors of this paper would like to acknowledge the support of Pranav Vyas* for assisting in L^AT_EX documentation and for advising in the context of Pre-Trained Transformers specifically for Sentiment Analysis and Similar NLP Tasks in Spanish. * - pranav.vyas2020 [at] vitstudent.ac.in

References

- [1] R. Labadie-Tamayo, B. Chulvi, P. Rosso, Everybody hurts, sometimes. overview of hurtful humour at iberlef 2023: Detection of humour spreading prejudice in twitter, in: Procesamiento del Lenguaje Natural (SEPLN), volume 71, 2023.
- [2] Z. M. Farooqi, S. Ghosh, R. R. Shah, Leveraging transformers for hate speech detection in conversational code-mixed tweets, 2021. URL: <https://arxiv.org/abs/2112.09986>.

- [3] S. Cui, Y. Han, Y. Duan, Y. Li, S. Zhu, C. Song, A two-stage voting-boosting technique for ensemble learning in social network sentiment classification, *Entropy* 25 (2023). URL: <https://www.mdpi.com/1099-4300/25/4/555>. doi:10.3390/e25040555.
- [4] R. H. H. Aziz, N. Dimililer, Twitter sentiment analysis using an ensemble weighted majority vote classifier, in: 2020 International Conference on Advanced Science and Engineering (ICOASE), 2020, pp. 103–109. doi:10.1109/ICOASE51841.2020.9436590.
- [5] R. T. Mutanga, N. Naicker, O. O. Olugbara, Hate speech detection in twitter using transformer methods, *International Journal of Advanced Computer Science and Applications* 11 (2020). URL: <http://dx.doi.org/10.14569/IJACSA.2020.0110972>. doi:10.14569/IJACSA.2020.0110972.
- [6] O. Sharif, M. M. Hoque, Tackling cyber-aggression: Identification and fine-grained categorization of aggressive texts on social media using weighted ensemble of transformers, *Neurocomputing* 490 (2022) 462–481. URL: <https://www.sciencedirect.com/science/article/pii/S0925231221018567>. doi:<https://doi.org/10.1016/j.neucom.2021.12.022>.
- [7] S. Raschka, Mlxtend: Providing machine learning and data science utilities and extensions to python’s scientific computing stack, *Journal of Open Source Software* 3 (2018) 638. URL: <https://doi.org/10.21105/joss.00638>. doi:10.21105/joss.00638.
- [8] R. Labadie, B. Chulvi, P. Rosso, HURtful HUMour (HUHU): Detection of humour spreading prejudice in Twitter, 2023. URL: <https://doi.org/10.5281/zenodo.7967255>. doi:10.5281/zenodo.7967255.
- [9] J. Cañete, G. Chaperon, R. Fuentes, J.-H. Ho, H. Kang, J. Pérez, Spanish pre-trained bert model and evaluation data, in: PML4DC at ICLR 2020, 2020.
- [10] J. Devlin, M. Chang, K. Lee, K. Toutanova, BERT: pre-training of deep bidirectional transformers for language understanding, *CoRR abs/1810.04805* (2018). URL: <http://arxiv.org/abs/1810.04805>. arXiv:1810.04805.
- [11] J. M. Pérez, J. C. Giudici, F. Luque, pysentimiento: A python toolkit for sentiment analysis and socialnlp tasks, 2021. arXiv:2106.09462.
- [12] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, R. Soricut, ALBERT: A lite BERT for self-supervised learning of language representations, *CoRR abs/1909.11942* (2019). URL: <http://arxiv.org/abs/1909.11942>. arXiv:1909.11942.
- [13] J. Cañete, S. Donoso, F. Bravo-Marquez, A. Carvallo, V. Araujo, ALBETO and DistilBETO: Lightweight Spanish language models, in: Proceedings of the Thirteenth Language Resources and Evaluation Conference, European Language Resources Association, Marseille, France, 2022, pp. 4291–4298. URL: <https://aclanthology.org/2022.lrec-1.457>.
- [14] V. Sanh, L. Debut, J. Chaumond, T. Wolf, Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter, *ArXiv abs/1910.01108* (2019).
- [15] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, Roberta: A robustly optimized BERT pretraining approach, *CoRR abs/1907.11692* (2019). URL: <http://arxiv.org/abs/1907.11692>. arXiv:1907.11692.
- [16] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, V. Stoyanov, Unsupervised cross-lingual representation learning at scale, *CoRR abs/1911.02116* (2019). URL: <http://arxiv.org/abs/1911.02116>. arXiv:1911.02116.
- [17] J. M. Pérez, D. A. Furman, L. Alonso Alemany, F. M. Luque, RoBERTuito: a pre-trained language model for social media text in Spanish, in: Proceedings of the Thirteenth Language Resources and Evaluation Conference, European Language Resources Association, Marseille, France, 2022, pp. 7235–7243. URL: <https://aclanthology.org/2022.lrec-1.785>.
- [18] M. García-Vega, M. Díaz-Galiano, M. García-Cumbreras, F. Del Arco, A. Montejo-Ráez, S. Jiménez-Zafra, E. Martínez Cámara, C. Aguilar, M. Cabezudo, L. Chiruzzo, et al., Overview

- of tass 2020: Introducing emotion detection, in: Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2020) Co-Located with 36th Conference of the Spanish Society for Natural Language Processing (SEPLN 2020), Málaga, Spain, 2020, pp. 163–170.
- [19] P. He, X. Liu, J. Gao, W. Chen, Deberta: Decoding-enhanced bert with disentangled attention, in: International Conference on Learning Representations, 2021. URL: <https://openreview.net/forum?id=XPZlaotutsD>.
- [20] Z. Yang, Z. Dai, Y. Yang, J. G. Carbonell, R. Salakhutdinov, Q. V. Le, Xlnet: Generalized autoregressive pretraining for language understanding, CoRR abs/1906.08237 (2019). URL: <http://arxiv.org/abs/1906.08237>. arXiv: 1906.08237.
- [21] K. Clark, M.-T. Luong, Q. V. Le, C. D. Manning, ELECTRA: Pre-training text encoders as discriminators rather than generators, in: ICLR, 2020. URL: <https://openreview.net/pdf?id=r1xMH1BtvB>.
- [22] L. Martin, B. Muller, P. J. O. Suárez, Y. Dupont, L. Romary, É. V. de la Clergerie, D. Seddah, B. Sagot, Camembert: a tasty french language model, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020.

Appendix

The Loss Function used in Subtask 1 (Binary Classification) and Subtask 2A (Multi-Label Binary Classification) are Variants of Cross-Entropy Loss which have been Described Below.

$$\mathcal{L}(y, \hat{y}) = -\frac{1}{N} \sum_{i=1}^N [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)] \quad (1)$$

$$\mathcal{L}(y, \hat{y}) = -\frac{1}{N * L} \sum_{i=1}^N \sum_{j=1}^L [y_{ij} \log(\hat{y}_{ij}) + (1 - y_{ij}) \log(1 - \hat{y}_{ij})] \quad (2)$$

The Loss used in Subtask 2B (Regression) can be computed using the Following Equation.

$$\mathcal{L}(y, \hat{y}) = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2 \quad (3)$$

The Equations for Descent using AdamW Optimizer have been Described Below.

$$\begin{aligned} m_t &= \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t, & v_t &= \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot g_t^2 \\ \hat{m}_t &= \frac{m_t}{1 - \beta_1^t}, & \hat{v}_t &= \frac{v_t}{1 - \beta_2^t} \\ \theta_t &= \theta_{t-1} - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon} \cdot (\hat{m}_t + \lambda \cdot \theta_{t-1}) \end{aligned} \quad (4)$$

In (1) and (2), y represents the true binary labels for N examples, where each example can have L labels. Similarly, \hat{y} represents the predicted probabilities for the corresponding labels. The loss is calculated by summing the cross-entropy loss for each label and averaging it over the number of examples. In (3), y represents the true target values, \hat{y} represents the predicted values, and N represents the number of examples. In (4), m_t and v_t represent the first and second moments (mean and uncentered variance) of the gradients respectively. β_1 and β_2 are the decay rates for the first and second moments. g_t represents the gradient at the current time step. \hat{m}_t and \hat{v}_t are the bias-corrected first and second moments. θ_t and θ_{t-1} represent the parameters at the current and previous time steps respectively. η is the learning rate, ϵ is a small value to prevent division by zero, and λ is the weight decay parameter.