Team Performance Pressure@SemEval 2022 Task 8: Evaluation of Multilingual News Article Similarity leveraging pretrained Models

Adarsh Kumar, Animesh Jain, Abhinav Bohra, Tanuj Saraf and Ishan Sharma
18MA20003, 18CS10004, 18CS30049, 18IE10039 and 18CS30020
{adarshkumar712, animesh99, abhinavbohra, saraft20}@iitkgp.ac.in
ishan996@iitkgp.ac.in

Abstract

In this paper, we describe our system entry for SemEval 2022 Task 8 which is on Multilingual News Article Similarity, where we leverage the knowledge of pre-trained language models to evaluate the Overall Similarity between a given pair of Articles. In our system, we use a Sentence transformer based approach to estimate the contextualized embeddings, on which we apply the Cosine similarity followed by a renormalisation, to get the final score. We further finetune the Model using the Cosine Similarity Loss (details of which is provided in Section 3) on the provided dataset. We also try to leverage the metadata provided with the Articles, by concatinating "Title" with the textual content, so as to improve the performance. We evaluate the model performance using the Pearson Correlation Score in both Multilingual and Translated to English settings. Our proposed approach using the Multilingual Setting is ranked 19th in the official SemEval 2022 Task 8 Leaderboard with a Pearson correlation score of **0.721**. In addition to our final approach, we also discuss some other approaches we experimented on, before arriving at our final model, in Section 4.3.

1 Introduction

The advent of news aggregators that collect thousands of articles from different publishers and websites has increased interest in detecting news stories connected to a specific theme and integrating them into news clusters. It's also been urgent and difficult to track the similarity of news coverage across different publications or locations. Monitoring and containment of infectious disease outbreaks, for example, has been a critical component of public health strategy to limit infections, whether earlier with Ebola or more recently with the COVID-19 pandemic. The ability to accurately detect disease outbreaks is crucial in the deployment of effective intervention strategies. As such reports may not only be in English, there is also a need for effective multilingual systems. Therefore, recent research has concentrated on detecting similarities between documents, phrases, and stories.

The SemEval 2022 Task 8 (Multilingual News Article Similarity) aimed at developing systems that identify multilingual news articles that provide similar information by rating them on a real-valued [1 – 4] scale, from most to least similar.

In this paper, we propose Sentence Transfomer based approach, followed by Cosine Similarity and renormalisation to evaluate the multilingual news article similarity on the dataset. The primary features of our approach as described in this paper are as follows:

- We leverage the knowledge of pretrained Multilingual language models to assess the similarity between pair of sentences.
- We finetuned the model using Cosine similarity Loss to attenuate the model parameters for improved Contexual Embeddings on the Given dataset.
- We additionally leverage the information from "Title" given in Metadata of Articles to improve the model performance.
- We demonstrate the efficacy of Cosine Similarity based score over the Regression based score.
- We illustrate the efficiency of our final approach against other approaches and further provide with an error analysis to explain different segments of our model's performance

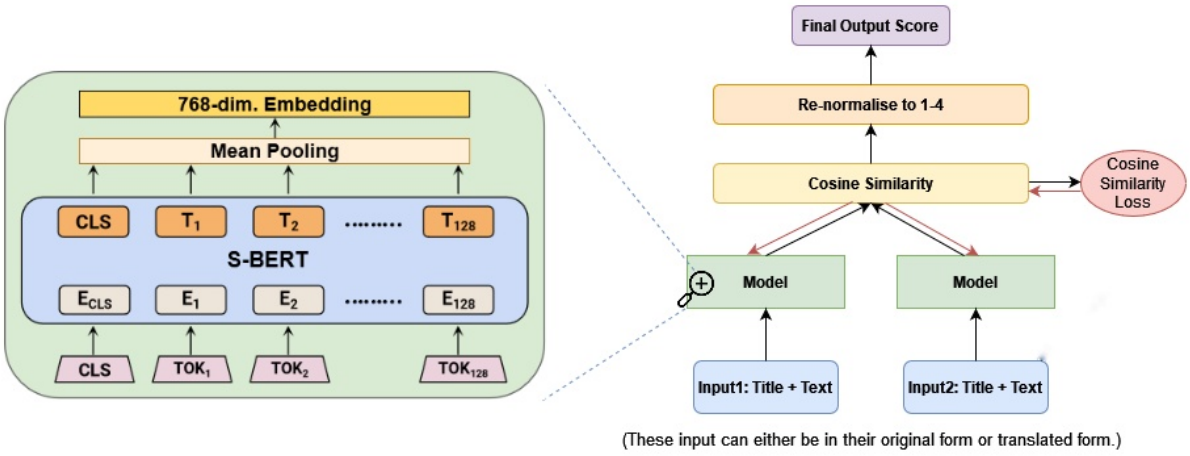

Figure 1: Model Architecture

2 Task and Dataset Description

The problem statement can be summarized as follows: "Given a pair of news articles, rate them on a 4-point scale from most to least similar".

The training dataset provided by the SemEval Task 8 consisted of 4,964[1] pairs of news articles (cross-lingual and monolingual pairs included) in the following languages: English(en), Spanish(es), German(de), Polish(pl), Turkish(tr), Arabic(ar) and French(fr). The news articles were provided as urls to the publisher's websites which were downloaded using a publicly available python script. Distribution of pairs is as follows: en-en: 1800, de-de: 857, de-en: 577, es-es: 570, tr-tr: 465, pl-pl: 349, ar-ar: 274, fr-fr: 72.

The evaluation data, consisting of 4954 pairs[2] of news articles, is given in a similar format without the additional information of Geography, Entities, time, Narrative, Overall, Style and Tone columns. It also contains news articles in unseen languages. Some more information on dataset can be seen in Section A.

The table below shows distribution of different types of language pairs across train, validation and test set. It is evident that there are more monolingual pairs in all the sets as compared to cross-lingual pairs. The three "surprise" languages that were not present in the training data are Chinese, Italian, and Russian.

[1]Out of these, 24 pairs were inaccessible while downloading, so we excluded those pairs for our assessment A

[2]Out of these, 243 pairs were inaccessible while downloading, so we excluded those pairs for our assessment A

Language Pair	Train Set	Val Set	Test Set
Monolingual	3603	784	3462
Cross-lingual	515	62	1440
Unseen pairs	NA	346	3131

Table 1: Counts of different types of language pairs

3 Approach

In this section we describe our final approach, which we used as our final submission on Codalab for the shared task. Our final approach consisted of two parts: Sentence Transformer(ST) Based Similarity and Finetuning ST with Cosine Similarity Loss. Our model architecture is depicted in the figure **??**. Before feeding into the model, we concatenated the "Title" along with Textual content for each article as input.

3.1 Sentence Transformer (ST) based Similarity

In this approach, we used the encoded embeddings from Sentence transformer (ST) model for each of the input text in a given pair. Using these embeddings, we evaluated the Cosine similarity score (between 0 to 1, from least to most), which was then renormalised to expected score (from 1 to 4, most to least).

3.2 Finetuning ST with Cosine Similarity Loss

In this approach, we finetuned the ST models using the cosine similarity loss between cosine similarity evaluated using Embeddings (as described in previous section) and the normalised target simi-

larity score. For normalisation, we converted the target score (gold-standard) from 1 to 4, to 1 to 0 (most similar to least similar in both cases). Once finetuned, we used the predicted cosine similarity score after renormalisation back to 1 to 4 as our Similarity score. Mathematically, the loss function used can be represented as follows:

$$MSE(y_{true_n m}, y_{pred}) = \sqrt{(\frac{1}{n})\sum_{i=1}^{n}(y_{true_n mi} - y_{pred_i})^2}$$

where $y_{true_n m}$ are the normalised gold-standard and y_{pred} being the predicted scores for similarity.

We experimented our final approach in two different settings: Multilingual and Translated. For the translated setting, we translated all the text and titles in articles to English using Google Translator[3]. We tried our approach with different SBERT[4](Reimers and Gurevych, 2019) and BERT models, including mBERT(Devlin et al., 2018) and XLM_R(Conneau et al., 2019). Results of both the settings are presented in the Section 5.

4 Experiments

4.1 Experimental Setup

All the experiments were performed using Pytorch (Paszke et al., 2019) and HuggingFace (Wolf et al., 2019) Transformers library. As the implementation environment, we used Kaggle Kernel which is a free research tool with a Nvidia Tesla P100 GPU and 16GB GPU Memory. Optimization was done using Adam (Kingma and Ba, 2017) with a learning rate of 5e-5. As discussed earlier in section 3, in our experiments, we used pre-trained mBert and XLM-Roberta models available in HuggingFace library. Input sentences were tokenized using respective tokenizers for each model, with maximum sequence length restricted to 512 tokens for mBert and 1024 for XLM-Roberta. We trained each regression model with a batch size of 16. In all the approaches, we used only the first token output provided by each model as input to regression layer. Each regression layer has 1 dropout layer with dropout of 0.2 and 1 fully connected layer.

4.2 Evaluation Metrics

For the evaluation, we have used weighted pearson correlation score as a metric for measuring the performance. Mathematically, pearson correlation score can be represented as:

$$r = \frac{\sum_{i=1}^{n}(y_{true_i} - \bar{y_{true}})(y_{pred_i} - \bar{y_{pred}})}{\sqrt{\sum_{i=1}^{n}(y_{true_i} - \bar{y_{true}})^2}\sqrt{\sum_{i=1}^{n}(y_{pred_i} - \bar{y_{pred}})^2}}$$

where y_{true} are the gold-standard and y_{pred} being the predicted renormalised scores for similarity (calculated after renormlisation of Cosine Similarity scores to 1 to 4, most to least similar).

4.3 Other Approaches

- **Baseline** consists of a pre-trained langauge model followed by regressor layer (i.e., a Dropout and Linear layer). The contexualized embeddings for each sentence of a given pair, obtained from model [CLS] token output (or pooled output) were concatenated before feeding into the regressor layer. We finetuned the entire model along with regressor layer using the MSE loss.

- **Baseline + Validation** uses a validation dataset constructed from train dataset. The final model used for evaluation in this case is the one that performs best on the validation dataset.

- **Baseline + Validation + Translation** uses Google Translate API to "normalise" two cross-lingual documents into a common language. These documents are then passed to Baseline+Validation model discussed earlier.

- **Baseline + Validation + "Title"** modifies input to mBERT/XLM-R as follows:

  $$\{t_1, t_2, \cdots, t_n; \langle SEP \rangle; d_1, d_2, \cdots, d_n\}$$

  where $t_1, t_2, \cdots, t_n$ are tokens from article's Title; $\langle SEP \rangle$ is the separator token; $d_1, d_2, \cdots, d_n$ are original document tokens.

- **Baseline + Validation + Losses**

  (a) **CosineEmbeddingloss** modifies the loss between predicted and target values as(margin 0.5):

  $$\text{loss}(x, y) = \begin{cases} 1 - \cos(x_1, x_2) & \text{if } y = 1 \\ \max(0, \cos(x_1, x_2) - \text{margin}) & \text{if } y = -1) \end{cases}$$

  (b) **Cosine Similarity + MSE Loss** Instead of feeding the concatenated contexual embeddings of two documents to

[3] https://pypi.org/project/googletrans/

[4] https://huggingface.co/sentence-transformers/paraphrase-multilingual-mpnet-base-v2

Regression			
Approach Category	**Approach**	**Model**	**Pearson Correlation**
Baseline	–	mBERT	0.3089
		XLM-R	0.2852
Baseline	+ Validation	mBERT	0.3164
		XLM-R	0.3246
Baseline + Validation	+ Translation	mBERT	0.3371
		XLM-R	0.3045
Baseline + Validation	+ Title	mBERT	0.3187
		XLM-R	0.3345
Baseline + Losses + Validation	+ CosineEmbeddingLoss	mBERT	0.3346
		XLM-R	0.3336
Baseline + Losses + Validation	+ Cosine Similarity + MSE Loss	mBERT	**0.3621**
		XLM-R	**0.3454**
Sentence Transformer	+ Translation + Title + MSE Loss	S-BERT	0.2864

Table 2: Results with Regressor layer (Dropout layer followed by a feed forward layer) for final prediction as described in Section 4.3. Note that here we feeding the concatenated contexualized embeddings from Model to the regressor layer

the regressor layer, compute the Cosine Similarity between embeddings vectors. Scale up the output to $[1, 4]$ and compute MSE loss with target.

5 Results and Discussion

The result from our experiments are provided in Table 2 and 3. From the tables we can note that after using combination of best approaches the pearson correlation score improved significantly (from 0.3621 to 0.6874). We can clearly see that our best model (Table 3) outperforms other approaches, by a significant margins. Further, the ablation study provided in Table 3, demonstrates the usefulness of different segments of our final approach in improving the score.

Furthermore, we can also observe that Similarity based Models perform way better than regression based Models. Also concatenating "Title" along with "Text" also show significant improvement from $\approx$ 0.6 to 0.7. Finetuning the model with Cosine Similarity further showed improvement in results from $\approx$ 0.70 to 0.75, which suggests that the model is able to refine the embeddings from the Sentence Transformer Model further with the pretraining and that this step works positively in our case (and there wasn't a case of catastrophic forgetting here).

Moreover, our final model stands at 19th place on Codalab leader-board for the SemEval Shared Task 8, with a pearson correlation score of **0.721** (refer Appendix A.1). Please note that this decrease is because for the final submission on Codalab, we assumed the unavailable documents (243 such pairs) as empty strings separated by [SEP] token (i.e. "[SEP]" is given as input in this case), which lead to a slight decrease in the performance on Codalab leader-board. However, during our own evaluations, we ignored such documents.

6 Error Analysis

Although we have received some interesting results, there were certain dimensions which didn't prove to work out positively in our case as was expected. As we can see from Results Table 2 and 3, similarity based methods significantly outperformed Regression ones. This behavior could be attributed to smaller size of data, which made it hard to train the Regressor layer, hence the low scores. We also tried out some other models like SVR etc, as the regressor layer, but the scores still didn't improve which seconds the point of small dataset for training regressor layer, against using Cosine Similarity.

Another point here worth analysing is that despite the fact we achieved a score of 0.7518 on Evaluation set, what kind of pairs our model performed poorer as compared to others. The same analysis is presented in Table 4. Interestingly, our model performed better on Cross-Lingual pairs as compared to the Monolingual Pairs. Despite the

Similarity		
Approach	**Model**	**Pearson Correlation**
ST Multilingual CS	S-BERT	0.5866
ST Translated CS	S-BERT	0.6058
ST Multilingual + "Title" CS	S-BERT	0.6778
ST Translated + "Title" CS	S-BERT	0.7086
**ST Multilingual + "Title" + CosSimLoss Finetuned CS	S-BERT	**0.7518**
**ST Translated + "Title" + CosSimLoss Finetuned CS	S-BERT	0.7492
ST(mBERT) + "Title" + Translation + CosSimLoss Finetuned CS	mBERT	0.6874
ST(XLM) +"Title" + Translation + CosSimLoss Finetuned CS	XLM-R	0.6357

Table 3: Results of the Consine Similarity based approach, where final score was evaluated using cosine similarity between embeddings and then scaling to 1 to 4. ** represents the models we used for our final submission on codalab as described in Section 3. Table also represent the ablation study for various elements included in our final approach along with our final approach experimented on mBERT in the ST framework. Note: **CS**: Cosine Similarity

Language Pair	Pearson Correlation
Monolingual	0.7386
Cross-lingual	0.7739
Unseen pairs	0.7459

Table 4: Pair-type wise Pearson Correlation score on Evaluation set

fact that our dataset consist a majority of Monolingual Sentences as compared to Cross-Lingual pairs, our model was able to learn differentiate between Cross Lingual pairs better. Moreover, a high pearson correlation score on unseen pairs combined suggest that because of our validation set quite similar to the Evaluation set, the case of catastrophic forgetting didn't arise. Finally, when we combine both of these observations, we can conclude that the case of over-fitting wasn't there.

Lastly, there were few more aspects of the data or model we experimented upon, but didn't improve the score much. Some of such experiments are presented in Appendix A. As we can see in Appendix A, concatenation of trained and no-trained embeddings didn't improve the score, which could again be attributed to the less data size here, similar to what we observed in regressor layer case. Further, no significant improvement in the Multitasking paradigm of modelling suggests that other similarity scores weren't much helpful in predicting the Overall scores, though we expected the otherwise. This problem could be specific to our data, as here other scores might not be as suggestive to predict overall similarity.

7 Path Not Taken

In this section, we mention some of the approaches we didn't adopted for our final submission, or didn't try out with their respective reasons:

- **Dropped Approaches:** We dropped certain approached in our final approach, as they didn't proved much useful in improving the scores of the task. This include, using Multitask based approach, Summarisation, concatenation with non finetuned model embeddings and few other approaches. Some of the results from these approaches are added in the appendix section, to describe what didn't work out in our case.
- **Didn't try (Time constraint):** Because of time constraint, we were not able to try out some approaches, which might have been a good experiments here. This include Adverserial Finetuning(Kumar et al., 2021), Hierarchical Modelling of Articles with LSTM(Wang and Fan, 2018) adding additional analysis from text like common word count between Monolingual pairs or in translated pairs. These approaches could be a great future work for our task, which we would definitely like to see, if they work out in this case or not.

8 Conclusion and Future Work

In this paper, we describe our approach to predict Similarity across Multilingual Article pairs provided in the SemEval Shared Task 8, leveraging the knowledge of pretrained Sentence Transformer

Based Models. As the evaluation results indicate, our final approach clearly outperforms other approaches we experimented on. Furthermore, our ablation study further suggests the importance of various elements in our final approach.

However, there is still a lot of scope of improvement in our approach, like as described in Section 7, second point. We believe that experimenting on those approaches might prove pretty useful in evaluation the final scores and are definitely worth trying in future.

9 Contribution Details

The contribution details of the members in our project are as follows:

Adarsh Kumar came up with almost all the approaches. He contributed in the literature review and all experiments related to SBERT, including the final model. He did initial experiments with Cosine Similarity, which was the most important feature of our final model. He made the final submission on the CodaLab competition. At the end he also came up with the error analysis part of the project, and finalised the report with Animesh.

Animesh Jain made the vanilla finetuned baselines and performed different experiments on it including adding *Validation*, *Translation* and *Title*. He analysed the models and combined the best performing aspects (like Cosine Similarity Loss, Sentence Transformers, "Title" etc.) of different experiments, and took the score from ≈ 0.3 to 0.6, which helped us to consider Cosine Similarity in our final models.

Abhinav Bohra contributed in the initial vanilla finetuned baseline. He then experimented on the frozen transformer model and concatenation of finetuned and non finetuned embeddings. Finally he also contributed in the final data analysis and also helped in writing the report.

Ishan Sharma experimented on MultiTask Approach (Shared Parameter Training), which gave satisfactory results with top 3 Correlated parameters with Overall. He performed data analysis with Abhinav. He helped in the literature review and also in writing the report.

Tanuj Saraf experimented on adding different regressors on the finetuned baselines. He also worked on the initial data handling, and some other exploratory analysis related to data. He also contributed in writing the report.

Acknowledgments

We would like to thank our TAs Soumya Sharma and Aniruddha Roy for their reliable feedback and suggestions throughout the project. Their guidance really helped us, in arriving to our final approach. We would also like to thank Professor Pawan Goyal for giving us an opportunity to work on this problem. We learned a lot with this project.

References

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. *CoRR*, abs/1911.02116.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Diederik P. Kingma and Jimmy Ba. 2017. Adam: A method for stochastic optimization.

Adarsh Kumar, Ojasv Kamal, and Susmita Mazumdar. 2021. Adversities are all you need: Classification of self-reported breast cancer posts on Twitter using adversarial fine-tuning. In *Proceedings of the Sixth Social Media Mining for Health (#SMM4H) Workshop and Shared Task*, pages 112–114, Mexico City, Mexico. Association for Computational Linguistics.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. *CoRR*, abs/1912.01703.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Bo Wang and Binwen Fan. 2018. Attention-based hierarchical lstm model for document sentiment classification. *IOP Conference Series: Materials Science and Engineering*, 435:012051.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *CoRR*, abs/1910.03771.

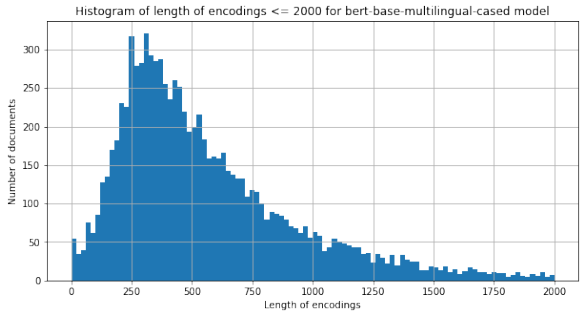

Figure 2: Token Length Distribution with mBert tokenizer

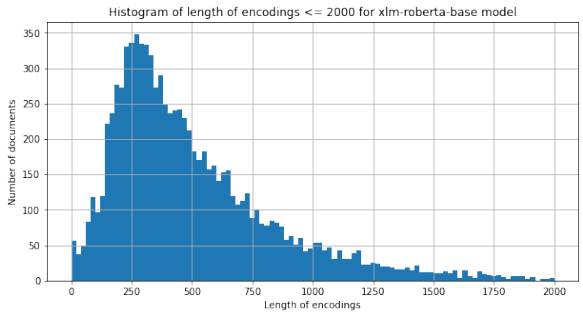

Figure 3: Token Length Distribution with XLM-Roberta tokenizer

A Appendix

Please note that we have ignored all those pairs for which atleast one of the files were inaccessible, and assessed our model for all the available files only.

Furthermore, the distribution of Token lengths across the dataset using tokenizers of mBert and XLM-Roberta are shown in Figures 2 and 3. Despite the fact that there were so many longer texts in the dataset, in our final approach of Concatenating Title with text, we used only 128 tokens as input. This was because the scores were lower for longer text inputs as compared to shorter ones.

A.1 CodaLab Submission Details

We had 243 and 24 pairs of files missing from the test set and train set respectively. So, for the final submission on CodaLab, we decided to fill the missing files as empty strings for both title and text (ie "[SEP]" only). Our submission details can be seen in Table 5. Submission results can be seen in Figure 4.

We have also secured **19th rank overall** in the leaderboard[5]. The screenshot of the leaderboard can be seen in Figure 5.

[5] https://competitions.codalab.org/competitions/33835#results

Submission Type	Submission Date (MM/DD/YY)	Final Correlation
Multilingual	04/02/22	0.721
Translated	04/02/22	0.717

Table 5: CodaLab Submission details

A.2 Further experiments

We performed some more experiments and explored different methods. As these experiments did not yield a very high pearson correlation score, we are listing them below in Table 6.

B Supplemental Material

Links to the Huggingface models used for our experiments:

- mBert: https://huggingface.co/bert-base-multilingual-cased
- XLM-Roberta: https://huggingface.co/xlm-roberta-base
- SBERT: https://huggingface.co/sentence-transformers/paraphrase-multilingual-mpnet-base-v2

Organization/affiliation:

IIT Kharagpur

Submit

Here are your submissions to date (✔ indicates submission on leaderboard):

#	SCORE	FILENAME	SUBMISSION DATE	STATUS	✔	
1	0.7205333427	Submission.zip	04/02/2022 07:52:23	Finished	✔	+
2	---	Submission_translated.zip	04/02/2022 08:02:42	Submitted		+
3	0.7172510638	Submission_translated.zip	04/02/2022 08:05:09	Finished		+

Figure 4: CodaLab Result page

17	*Anonymous*	3	01/30/22	Andi	0.726 (17)	View
18	*Anonymous*	4	01/27/22	BUT	0.726 (18)	View
19	*Anonymous*	3	04/02/22		0.721 (19)	View
20	*Anonymous*	3	02/01/22	TCU	0.715 (20)	View

Figure 5: CodaLab Leaderboard

Approach	**Model**	**Pearson Correlation**
Baseline + Validation + Frozen Transformer Model	mBERT XLM-R	0.0376 0.0910
Concat(finetuned, non-finetuned embeds) + Regressor Layer	mBERT XML-R	0.3048 0.2962
Multi-task(Shared Parameter) + Top 3 Correlated params with Overall	mBERT XLM-R	0.3100 0.2556
Multi-task(Shared Parameter) + All 7 extra fields	mBERT XLM-R	0.2830 0.2662

Table 6: Other Experiments