Multi Output Learning using Task Wise Attention for Predicting Binary Properties of Tweets : Shared-Task-On-Fighting the COVID-19 Infodemic

Ayush Suhane*
IIT Kharagpur
Kharagpur, India
ayushsuhane99@iitkgp.ac.in

Shreyas Kowshik*
IIT Kharagpur
Kharagpur, India
shreyaskowshik@iitkgp.ac.in

Abstract

In this paper, we describe our system for the shared task on Fighting the COVID-19 Infodemic in the English Language. Our proposed architecture consists of a multi-output classification model for the seven tasks, with a task-wise multi-head attention layer for inter-task information aggregation. This was built on top of the Bidirectional Encoder Representations obtained from the RoBERTa Transformer. Our team, dunder_mifflin, was able to achieve a mean F1 score of 0.891 on the test data, leading us to the second position on the test-set leaderboard.

1 Introduction

In recent years, the spread of misinformation on social media has been growing rapidly. Amid the global pandemic, the spread of misinformation has had serious consequences. Covid-19 misinformation caused mass hysteria and panic; reluctance to use masks and follow social distancing norms; denial of the existence of Covid-19, anti-vaxxers, etc. There is a need for automatic detection and flagging of tweets spreading misinformation.

Automatic detection of misinformation is an open research problem in NLP. Misinformation is intentionally written to deceive and pass as factual which makes detecting it a difficult task.

(Silva et al., 2020) analyzed a dataset of 505k tweets. They used sentiment analysis, polarity scores, and LIWC to build features. They used ML models such as RandomForest, AdaBoost, SVM, etc to classify tweets as factual/misinformation.

Predicting answers to multiple questions, which is the setup of our current problem statement, can be modeled as a multi-task-learning problem. (Crawshaw, 2020) have highlighted different methods for sharing information among tasks, for task-specific performance boosts, such as cross-stitching and soft-parameter-sharing. They also highlight ways for loss weighting based on task-dependent uncertainty and learning-speed.

(Liu et al., 2019) have highlighted the use of attention mechanisms for the multi-task learning problem and show that it performs competitively with other approaches.

Inspired by this idea, we propose an Attention-Based architecture **RoBERTa Multihead Attn** for our task by incorporating inter-task information for task-specific performance enhancement. With a test-set F1 score of **0.891**, our approach shows the superiority of combining information among tasks over modeling them independently and shows the effectiveness of multihead-attention for this purpose.

2 Task Description and Dataset

The objective of this task (Shaar et al., 2021) is to predict various binary properties of tweets. We were given several tweets where we had to predict answers to 7 questions. These questions pertain to whether the tweet is harmful, whether it contains a verifiable claim, whether it may be of interest to the general public, whether it appears to contain false information, etc. There were three tracks of languages on English, Arabic, and Bulgarian and a team was free to choose any subset of the languages.

For the English Language, the training dataset consisted of 862 tweets [1] The dev set consisted of 53 tweets and the testing set consisted of 418. The training dataset statistics are shown in Table 1.

* Denotes Equal Contribution

[1] In the training dataset (869 tweets) provided by the organizers, 7 tweets were mislabeled and had "Nan" in Q6 or Q7 (which only have classes "yes" and "no"). As instructed by the organizers, we dropped those tweets from the dataset and were left with 862 tweets, which were used for further analysis.

Proceedings of the 4th NLP4IF Workshop on NLP for Internet Freedom, pages 110–114

Question-Id	No	Yes	Nan
Q1	298	564	-
Q2	457	39	366
Q3	50	506	306
Q4	407	153	302
Q5	383	181	298
Q6	726	136	-
Q7	634	228	-

Table 1: Training dataset statistics

3 Methodology

We explored different base embeddings inspired by (Devlin et al., 2019) and (Liu et al., 2020). These are state-of-the-art language models which when used with task-specific fine-tuning, perform well on a wide variety of tasks. The embeddings are passed to our task-specific architecture for further processing and the whole model is trained end-to-end.

We hypothesize that the prediction for one question may benefit from the use of information needed to predict another question. For instance, questions Q4 ("Harmfulness: To what extent is the tweet harmful to the society/person/company/product?") and Q6 ("Harmful to Society: Is the tweet harmful to the society and why?") in the task have a deduction process that may have several common steps. To model this, we add an inter-task multi-head attention layer before our prediction heads.

3.1 Preprocessing

Before feeding the tweets to the RoBERTa tokenizer, we performed the following operations:

1. We observed that there were a lot of non-ASCII characters in the tweets, so we stripped these characters.

2. We then replaced the emojis with their text description using the demoji python package.

3. We replaced all the links in the tweets with "URL" and mentions with "USER" tokens.

3.2 Data Augmentation

Due to the small size of the dataset, we used data augmentation to improve generalization. Back-Translation was used for this purpose. Given an input, the text is first translated into a destination language to obtain a new sentence. This new sentence is then translated back into the source language. This process creates a slightly modified version of the original sentence, while still preserving the original meaning. We carried this out using 3 destination languages (French, Spanish, German) using the MarianMT (Neural Machine Translation Model)[2]. As an example :

Original Tweet

For the average American the best way to say if you have covid-19 is coughing in a rich person face and waiting for their test results

Augmented Tweet

For the average American the best way to tell if you have covid-19 is to cough in a rich person's face and wait for their test results

3.3 Task-Wise Multi-head Attention Architecture

Multi-Head-Attention (Vaswani et al., 2017) has shown to capture representations across different subspaces, thus being more diverse in its modeling compared to Single-Head-Attention. Inspired by this, we added a multi-head-attention layer to aggregate information among different tasks.

Our entire architecture is shown in Figure 1. The input sentences are encoded using the RoBERTa tokenizer. These are then forward propagated to get the embedding vectors. This vector is passed through a linear-block[3] and then branched out using 7 different linear layers, one for each task. These are further processed to obtain the 7 task-specific vectors (we will refer to these as task vectors).

Each of these vectors is then passed through a multi-head-attention layer with the vector itself being the query vector, and the concatenated task vectors being the key and value vectors. The attention weights captured through this method signify what proportion of information the model would want to propagate further from each of the task-specific vectors. The information from all the task vectors is thus aggregated as their weighted sum to get the penultimate layers for each task. Note that the projection matrices for multihead-attention for all tasks are independent of each other. A final linear layer maps these to the prediction layers on which softmax is applied for per-task prediction.

[2]MarianMT

[3]Linear-Block is a linear-layer+ReLU

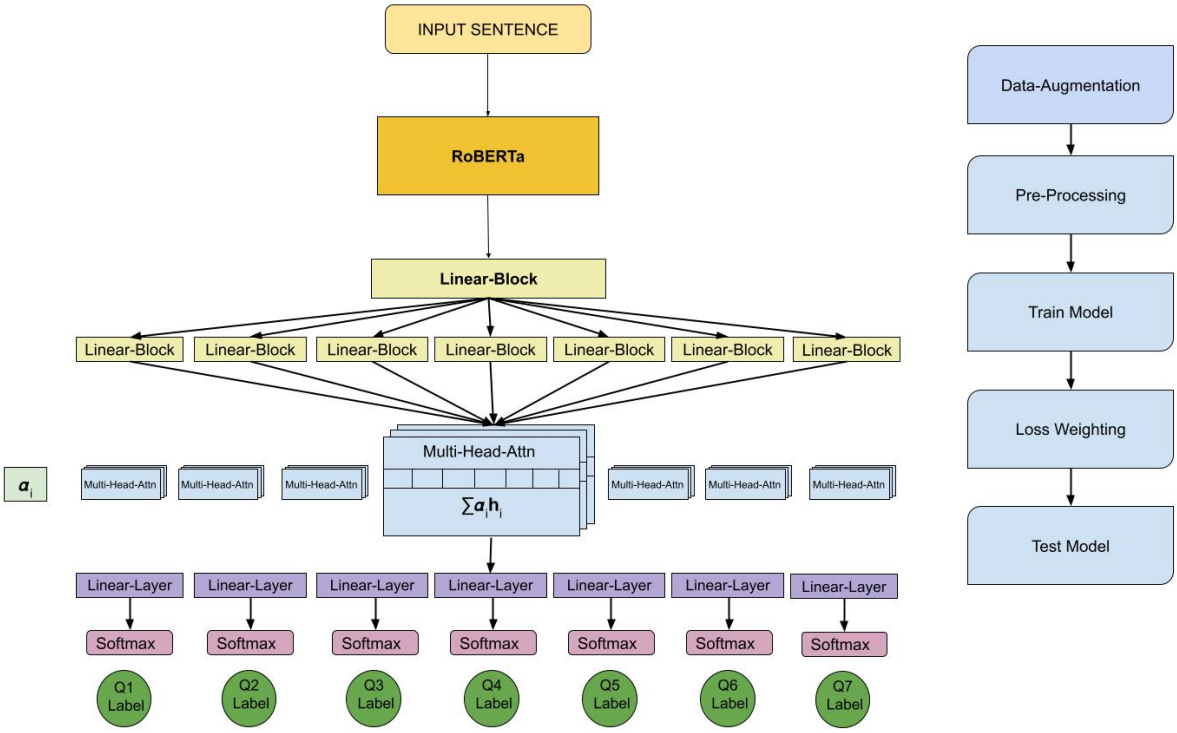

Figure 1: (Towards Left) Our proposed architecture for Roberta-Multihead-Attention. h_i denotes the embedding after projecting the value vector (the i^{th} task vector) using the multihead-attention layer's value-projection matrix. α_i denotes the attention weight for i^{th} task vector, which is the amount of information to propagate forward from that task vector. (Towards Right) Preprocessing and training pipeline

Model	Mean F1	F1 Q1	F1 Q2	F1 Q3	F1 Q4	F1 Q5	F1 Q6	F1 Q7
ngram_baseline	0.828	0.647	0.904	0.992	0.761	0.800	0.873	0.821
majority_baseline	0.830	0.612	**0.927**	**1.000**	0.770	0.807	0.873	0.821
RoBERTa Multihead Attn	**0.891**	**0.807**	0.923	0.966	**0.868**	**0.852**	**0.940**	**0.884**

Table 2: Evaluation results on official test set. Baselines provided by the organizers

Model	Mean F1
Vanilla BERT	0.786
BERT Multihead Attn	0.812
RoBERTa Multihead Attn	**0.823**

Table 3: Evaluation results on official dev set.

3.4 Loss Weighting

The input data is skewed in the distribution of labels for each question. A natural approach to tackle this issue is to use a weighted loss. For each task, we use the following scheme for assigning weights :

$$w_c = \frac{N_{samples}}{N_{classes} * N_c}$$

where $N_{samples}$ is the number of input data samples, $N_{classes}$ is the number of classes and N_c is the number of samples for class c for a particular task. This weighting was done independently for each task, based on the label distribution for that particular task.

4 Experiments

We conducted our experiments with $\text{Bert}_{\text{BASE}}$ and $\text{Roberta}_{\text{BASE}}$. All our code is open-source and available on Github [4].
The different architectures are explained below:

- **Vanilla BERT** : The input sentence is passed through the $\text{BERT}_{\text{BASE-UNCASED}}$ model. The output is first processed through a couple of linear blocks and finally branched out for task-wise linear layers for predictions for each of the 7 questions.

[4] https://github.com/shreyas-kowshik/nlp4if

Model	Language	Mean F1
Vanilla BERT	Arabic	0.706
BERT Multihead Attn	Arabic	**0.726**
Vanilla BERT	Bulgarian	0.819
BERT Multihead Attn	Bulgarian	**0.825**

Table 4: Evaluation results on official dev set for Arabic and Bulgarian

- **Bert Multihead Attn** : Figure 1 shows this architecture, with RoBERTa replaced by BERT. The input sentence is passed through the BERT transformer to obtain the bidirectional encoder representations. These are passed through a couple of linear-blocks and then branched out to 7 linear layers, each one corresponding to a task. For each branch, the output from the seven linear layers is then fed into a separate multi-head attention layer, with 3 heads. The output from the multihead attention for each task is finally passed through a linear layer and softmax and is used for predictions.

- **RoBERTa Multihead Attn** : This model is the same as Bert Multihead Attn except that the transformer used is $\text{RoBERTa}_{\text{BASE}}$.

All models were finetuned end-to-end, with weights also being updated for the embedding layers of BERT and RoBERTa. The loss function was weighted-cross-entropy for each task, and the final loss was the sum of losses for the 7 tasks. The learning rates followed a linear decay schedule starting from an initial learning rate. The models were trained in PyTorch with the HuggingFace library (Wolf et al., 2020).

Regarding the results in Table 3, we see that RoBERTa Multihead Attn performs the best overall on the development set. We obtain a significant boost in performance, over Vanilla-BERT, by using our proposed Multihead-Attention layers. Using RoBERTa embeddings further brings about a slight improvement over this. We thus finalize Roberta Multihead Attn as our final model for submission.

For this particular experiment, we used a learning rate of 5e-5 for the task-specific layer and 5e-6 for the RoBERTa fine-tuning layers. The model was trained for 60 epochs with the number of attention-heads set to 3. All layers except the penultimate, attention, and RoBERTa layers had a 0.1 dropout probability.

Roberta Multihead Attn beats the baselines by a significant margin overall as shown in Table 2 and ranks 2nd on the test-set leaderboard for the English Language sub-task, with a test-set mean F1-Score of 0.891.

Upon request from the reviewers, we also show the results on the development set for the given architecture on the Arabic and Bulgarian datasets. The bert-base-multilingual-cased embeddings were used as the base embeddings for these experiments. With reference to Table 4 we see that our proposed architecture outperforms the Vanilla BERT architecture on both the Arabic and Bulgarian datasets, further illustrating it's effectiveness across languages.

5 Conclusion and Future Work

In this paper, we have described our system for predicting different binary properties of a tweet, on the English sub-task of the Shared Task On Fighting the Covid Infodemic in NLP4IF'21. Our approach uses the $\text{RoBERTa}_{\text{BASE}}$ architecture for the initial embeddings and builds on top of that using task-wise multi-head attention layers. Our results show that using a multi-head attention approach for aggregating information from different tasks leads to an overall improvement in performance.

Possible developments in this task can include the incorporation of additional contextual information in our models using tweet-related features like the #(number) of URLs in a tweet, the # of user mentions and the # of hashtags, etc. Also, user-related features such as the # of followers, account age, account type (verified or not) can be included. These features contain a lot of auxiliary information that can aid in fine-grained classification. Sociolinguistic Analysis such as Linguistic Inquiry and Word Count (LIWC) can also be used to gather emotional and cognitive components from the tweet.

6 Acknowledgements

We would like to thank everyone in the organizing committee and the reviewers for reviewing our paper and providing their valuable feedback. We would also like to thank Chinmay Singh and Prakhar Sharma for proof-reading the paper and providing other valuable suggestions towards improving the paper.

References

Michael Crawshaw. 2020. Multi-Task Learning with Deep Neural Networks: A Survey. *arXiv e-prints*, page arXiv:2009.09796.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Shikun Liu, Edward Johns, and Andrew J. Davison. 2019. End-to-end multi-task learning with attention. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1871–1880.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Ro{bert}a: A robustly optimized {bert} pretraining approach.

Shaden Shaar, Firoj Alam, Giovanni Da San Martino, Alex Nikolov, Wajdi Zaghouani, Preslav Nakov, and Anna Feldman. 2021. Findings of the NLP4IF-2021 shared task on fighting the COVID-19 infodemic and censorship detection. In *Proceedings of the Fourth Workshop on Natural Language Processing for Internet Freedom: Censorship, Disinformation, and Propaganda*, NLP4IF@NAACL' 21, Online. Association for Computational Linguistics.

Mirela Silva, Fabrício Ceschin, Prakash Shrestha, Christopher Brant, Juliana Fernandes, Catia Silva, André Grégio, Daniela Oliveira, and Luiz Giovanini. 2020. Predicting misinformation and engagement in covid-19 twitter discourse in the first months of the outbreak.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. pages 38–45.