AIR-JPMC@SMM4H'22: BERT + Ensembling = Too Cool: Using Multiple BERT Models Together for Various COVID-19 Tweet Identification Tasks

Leung Wai Liu, Akshat Gupta, Saheed O. Obitayo, Xiaomo Liu, Sameena Shah
J.P. Morgan AI Research, New York, NY, USA
leungwai@wustl.edu, {akshat.x.gupta, saheed.o.obitayo,
xiaomo.liu, sameena.shah}@jpmchase.com

Abstract

This paper presents my submission for Tasks 1 and 2 for the Social Media Mining of Health (SMM4H) 2022 Shared Tasks competition. I first describe the background behind each of these tasks, followed by the descriptions of the various subtasks of Tasks 1 and 2, then present the methodology. Through model ensembling, this methodology was able to achieve higher results than the mean and median of the competition for the classification tasks.

1 Introduction

Social media has grown to be an essential communication platform. Massive platforms like Facebook have billions of users (Heath, 2022), enabling it to gauge public sentiment about a plethora of topics, two of them being pharmacovigilance and health mandates. Pharmacovigilance is the assessment of adverse effects from medical drugs (WHO, 2022). In the past, pharmacovigilance information were gathered from marketing clinical trials, or post-marketing reporting by physicians, which pose limitations on information quantity. (O'Connor et al., 2014). Platforms like Twitter are also a key tool to gauge sentiment about public events, such as health mandates during a public health crisis. Understanding public sentiment is pertinent to get everyone on the same page.

The issue with Twitter data, however, is its lack of uniformity, making it hard to automate sentiment classification. This is where Natural Language Processing (NLP) and this competition comes into play. Hosting for the seventh time (Weissenbacher et al., 2022), this Shared Tasks competition, like in previous years, tackles many social media data related tasks, including pharmacovigilance (Magge et al., 2021a). This system builds upon state-of-the-art models to further improve results for classification tasks.

1.1 Task Description

Task 1 is a pharmacoviligiance task consisting of three subtasks, building upon one another:

- **Task 1a - ADE Classification:** Classify whether a tweet is about an Adverse Event of a drug (**ADE**) or not (**noADE**).
- **Task 1b - ADE Span Detection:** Given an **ADE** classified tweet, detect the span that pertains to such ADE.
- **Task 1c - ADE Entity Normalization:** Given an **ADE** span prediction, map that to its respective MedDRA concept ID Label.

Task 2 is a health mandate related task. Given three health mandate labels: **face masks**, **stay at home orders**, and **school closures**, the two subtasks are:

- **Task 2a - Stance Detection:** Classify whether the tweet of a particular label is in **FAVOR**, **AGAINST**, or **NONE** opinion of the mandate.
- **Task 2b - Premise Classification:** Detect whether the tweet has a premise pertaining to the label (labeled **1**), or not (labeled **0**).

Tasks 1a, 2a, and 2b are classification tasks, sharing similar methodology, while Tasks 1b (span detection), and 1c (entity normalization) take a different approach.

2 Datasets

The dataset used for Task 1 is Version 2 of the DeepADEMiner dataset from Magge et al. (2021b), while the dataset used for Task 2 is the stance detection dataset from Davydova and Tutubalina (2022). The training and validation sets have labels for training and validation, respectively, while the testing sets did not. Tables 1 and 2 shows the distribution of tweets for Tasks 1 and 2, respectively.

Proceedings of the 29th International Conference on Computational Linguistic, pages 163–167
October 12–17, 2022.

Dataset	ADE	noADE	Total
Training	1214	15960	17174
Validation	65	844	909
Test	—	—	10969

Table 1: Distribution of Task 1's ADE and noADE classes. Since the test set is unlabeled, the distribution is unknown.

Dataset	AGAINST	FAVOR	NONE	1	0	Total
Training	874	1346	1336	1331	2225	3556
Validation	158	244	198	220	380	600
Test	—	—	—	—	—	2000

Table 2: Distribution of Task 2's stance and premise classes. Since the test set is unlabeled, the distribution is unknown.

Task 1's distribution of labels are unbalanced, skewing towards more **noADE** class tweets. Task 2 has roughly equal distribution across its stance and premise classes.

3 Methodology

3.1 Pre-Processing

No pre-processing was performed besides merging the dataset file of the Tweet IDs and its labels together for training and validation for the classification tasks. For Task 1b, a different transformer model from the HuggingFace library was used (BERTForTokenClassification instead of AutoModelForSequenceClassification), and the label tokens were set up as 0/1/2: 0 being an irrelevant word in the sentence, 1 being the relevant span in the sentence, and 2 being padding for outside the sentence. This approach was adapted from Batcha (2021)'s Kaggle guide.

3.2 Models Used

For all tasks, the BERT language model (Devlin et al., 2019) was used, as it is a state-of-the-art language model that can be fine-tuned for many different tasks. The classification tasks were also trained with the RoBERTa language model (Liu et al., 2019), which improves upon BERT with tweaks in its training method. **BERT$_{\text{BASE}}$-uncased** and **RoBERTa$_{\text{BASE}}$** were used to get a baseline performance metric. After that, the datasets were trained on **BERT$_{\text{LARGE}}$-uncased** and **RoBERTa$_{\text{LARGE}}$**, as the larger models improve on accuracy than the base models.

The base models were trained on the training datasets for 5 iterations and 15 epochs per iteration across all classification tasks. The large models were trained on the training datasets for 3 iterations for Task 1a and 5 iterations for Tasks 2a and 2b, all with 15 epochs per iteration.

The methods to calculate the F1, precision, and recall metrics were different across subtasks as well. Task 1a requires the F1-score, precision, and recall for the **ADE** class only, while Task 2 requires the macro F1 score for the **AGAINST** and **FAVOR** classes only for Task 2a, and for both classes **1** (premise) and **0** (no premise) for Task 2b. In addition, Task 2 uses this formula to calculate the F1-score: $F_{1_{total}} = \frac{1}{n} * \sum_{c \in C} F_{1_{relc}}$, with C being the different labels (**face masks**, **stay at home orders**, **school closures**), and $n = 3$ being the size of set C. This means that after generating the predictions of the validation set, the results were split by label to find their individual F1 scores first, then combined together to find the total F1-score. Table 3 shows the baseline results of the F1, precision and recall of the classification tasks.

For Task 1b, only **BERT$_{\text{LARGE}}$-uncased** was used, as specific parsing of its tokens was required during post-processing. The model was trained on the training datasets for 5 iterations and 20 epochs per iteration, keeping the last epoch per iteration. No metrics were measured for Tasks 1b and 1c due to time constraints.

F1 / Task	BERT-base-uncased	RoBERTa-base	BERT-large-uncased	RoBERTa-large
Task 1a	0.587	0.667	0.742*	0.803*
Task 2a	0.629	0.665	0.682	0.746
Task 2b	0.765	0.769	0.651	0.647

Table 3: Performance of BERT and RoBERTa models on validation data for all classification tasks. The F1 scores is the average F1-score among all five models for each category (with the exception of certain Task 1a results marked with a (*), which takes the average of 3 models).

3.3 Span Detection Post-Processing

Although the classification tasks did not require post-processing before model ensembling, it was necessary for the span detection task to perform post-processing, to convert the predicted tokens to a predicted span string with beginning and end indices for where it is in the original sentence. Here is an explanation of how the post-processing code works:

1. Convert the original and predicted token ids

into word tokens.

2. If predicted span list IS empty, then predicted span string = "", begin and end indices = 0.
3. ELSE if the predicted span list is NOT empty:
   (a) Process original tweet, removing all punctuation and capitalization.
   (b) Retrieve the first and last word in predicted span list
   (c) Search for that word (removing ## if it is a partial word) in the full processed tweet using `regex`. This returns the beginning of the span.
   (d) If the beginning of the span is empty, then start index = 0, otherwise retrieve start index.
   (e) If predicted span list has ONLY one token: retrieve end index of that begin span.
   (f) ELSE if predicted token token list has more than one token:
      i. Reduce the processed full tweet to start at the begin span's start index.
      ii. Find first word that matches token (removing ## if it is a partial word) , then retrieve end index of that span.

Once a predicted span string, begin and end indices have been generated, model ensembling occurs.

3.4 Model Ensembling

After the models have been trained, a series of ensembling methods were conducted to improve the accuracy of the predictions. This includes: majority-vote, unweighted average, and weighted average of all 10 large models, and separately by model. The unweighted average was calculated with this equation: $\lfloor(\Sigma x)/n\rceil$, with x being the prediction and n being the number of models used in the ensemble, rounded to the nearest digit. The weighted average was calculated with this equation: $\lfloor(\Sigma xf)/c\rceil$, with f being the F1 score of the particular model, and c being the sum of all the models' F1-score used in the ensemble, rounded to the nearest digit.

Ultimately, a majority ensemble using **RoBERTa$_{\text{LARGE}}$** models only was used for Tasks 1a and 2a, while a weighted average ensemble with **BERT$_{\text{LARGE}}$-uncased** models was used for Task 2b. Table 4 shows the F1, precision, and recall metrics after ensembling for the classification tasks.

Metric / Task	F1-Score	Precision	Recall
Task 1a	0.838	0.803	0.877
Task 2a	0.771	—	—
Task 2b	0.816	—	—

Table 4: Performance metric on model ensembles for classification tasks. NOTE: Tasks 2a and 2b does not have the overall precision and recall scores calculated, as only the overall F1-score is used.

For Task 1b, the span with the median span length out of all 5 models is chosen. This enables the filling an answer for any tweets a particular model was unable to predict. If after ensembling there is no prediction, the entire tweet will be used as the prediction, as it helps with the overlapping performance metric.

The model ensembling method is used because multiple models generating a prediction together is proven to be effective in further increasing accuracy of predictions compared to a singular model (Jayanthi and Gupta, 2021).

3.5 Task 1c Post-Processing

Due to time constraints, entity normalization was not performed for Task 1c. Using the predicted span strings from Task 1b, the NLTK framework was used to remove the stop words from the span. Then, each remaining word in the span is searched in the MedDRA library, and the first match found for the span is submitted as the prediction.

4 Results

Table 5 on the next page shows the metrics on test data for Tasks 1 and 2. The classification tasks performed better than the mean and median scores of the competition across all metrics.

However, Task 1b (span prediction) performed at or above the mean in overlapping metrics while worse in strict metrics. This concurs with the submitted system, as it only takes the start index of the first word and last index of the last word in the predicted token list to build the predicted span, meaning that any gaps in between the predicted tokens are included in the final span. This was done to account for multiple spans in the same tweet in the training data.

For Task 1c (entity normalization), the first-term approach, as expected, performed worse than the mean metrics across the board.

TASK 1						
	Mean F1	**Submitted F1**	**Mean Precision**	**Submitted Precision**	**Mean Recall**	**Submitted Recall**
1a	0.562	**0.693**	0.646	**0.772**	0.497	**0.629**
1b	0.527 **(0.341)**	**0.568** (0.091)	0.539 **(0.344)**	**0.671** (0.114)	**0.517 (0.339)**	0.492 (0.076)
1c	**0.116 (0.083)**	0.070 (0.011)	**0.120 (0.085)**	0.087 (0.014)	**0.112 (0.082)**	0.058 (0.009)

TASK 2			
	Mean F1-score	**Median F1-score**	**Submitted F1-score**
2a	0.491	0.550	**0.577**
2b	0.574	0.647	**0.701**

Table 5: Performance Metrics on Test Data for Tasks 1 and 2. NOTE: For Tasks 1b and 1c, the values on the left are overlapping metrics while the values in the right in parenthesis are strict metrics. The values in bold are the higher value in the specific category and/or subtask.

5 Conclusion

As shown in the results section, better results were achieved than the baseline through model ensembling. This resulted in our system performing better than the mean and median metrics of the competition for all of the classification tasks and on-par with the overlapping metrics for Task 1b.

Further work can be done to improve this system. For the classification tasks, pre-processing of the dataset, such as removing extraneous punctuation or hashtags, may generate better results. For Task 1b, keeping the best epoch based on F1-score in addition to getting rid of the gaps in the span prediction may help especially in the strict metrics. Lastly, having the opportunity to conduct entity normalization properly may help improve Task 1c metrics as well.

References

Thanish Batcha. 2021. Bert for Token Classification (NER) - Tutorial.

Vera Davydova and Elena Tutubalina. 2022. SMM4H 2022 Task 2: Dataset for stance and premise detection in tweets about health mandates related to COVID-19. In *Proceedings of the Seventh Social Media Mining for Health (#SMM4H) Workshop and Shared Task*, pages –.

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.

Alex Heath. 2022. Facebook lost daily users for the first time ever last quarter.

Sai Muralidhar Jayanthi and Akshat Gupta. 2021. SJ_AJ@DravidianLangTech-EACL2021: Task-Adaptive Pre-Training of Multilingual BERT models for Offensive Language Identification. In *Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages*, pages 307–312, Kyiv. Association for Computational Linguistics.

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.

Arjun Magge, Ari Klein, Antonio Miranda-Escalada, Mohammed Ali Al-Garadi, Ilseyar Alimova, Zulfat Miftahutdinov, Eulalia Farre, Salvador Lima López, Ivan Flores, Karen O'Connor, Davy Weissenbacher, Elena Tutubalina, Abeed Sarker, Juan Banda, Martin Krallinger, and Graciela Gonzalez-Hernandez. 2021a. Overview of the Sixth Social Media Mining for Health Applications (#SMM4H) shared tasks at NAACL 2021. In *Proceedings of the Sixth Social Media Mining for Health (#SMM4H) Workshop and Shared Task*, pages 21–32, Mexico City, Mexico. Association for Computational Linguistics.

Arjun Magge, Elena Tutubalina, Zulfat Miftahutdinov, Ilseyar Alimova, Anne Dirkson, Suzan Verberne, Davy Weissenbacher, and Graciela Gonzalez-Hernandez. 2021b. DeepADEMiner: a deep learning pharmacovigilance pipeline for extraction and normalization of adverse drug event mentions on Twitter. In *Journal of the American Medical Informatics Association*, pages 2184–2192. American Medical Informatics Association.

Karen O'Connor, Pranoti Pimpalkhute, Azadeh Nikfarjam, Rachel Ginn, Karen L Smith, and Graciela Gonzalez. 2014. Pharmacovigilance on Twitter? Mining Tweets for Adverse Drug Reactions. In *Pharmacovigilance on Twitter? Mining Tweets for Ad-*

verse Drug Reactions, pages 924–933, United States. American Medical Informatics Association.

Davy Weissenbacher, Ari Z. Klein, Luis Gascó, Darryl Estrada-Zavala, Martin Krallinger, Yuting Guo, Yao Ge, Abeed Sarker, Ana Lucia Schmidt, Raul Rodriguez-Esteban, Mathias Leddin, Arjun Magge, Juan M. Banda, Vera Davydova, Elena Tutubalina, and Graciela Gonzalez-Hernandez. 2022. Overview of the Seventh Social Media Mining for Health Applications (#SMM4H) shared tasks at COLING 2022. In *Proceedings of the Seventh Social Media Mining for Health (#SMM4H) Workshop and Shared Task*, pages –.

WHO. 2022. What is Pharmacovigilance?