Team ITTC at BioCreative VII LitCovid Track 5: combining pre-trained and bag-of-words models

Yulia Otmakhova
University of Melbourne
Melbourne, Australia
yotmakhova@student.unimelb.edu.au

Antonio Jimeno Yepes
RMIT University / University of Melbourne
Melbourne, Australia
antonio.jose.jimeno.yepes@rmit.edu.au

***Abstract*—In this report we present the results of our experiments for the BioCreative VII Track 5 challenge organized by the US NIH / National Library of Medicine. The task, based on a sample from a manually curated database of COVID-19 studies [3], is to automatically assign one or more out of 8 topic labels (such as *Diagnosis*, *Treatment*, *Transmission*) based on an article's abstract, title and metadata. We have evaluated several machine learning methods and the best result was obtained by combining a bag-of-words approach based on Support Vector Machines and a BERT based model pre-trained in the biomedical domain.**

***Index Terms*—COVID-19, BioCreative, indexing, text categorization, machine learning**

I. Introduction

In this report we present the results of our experiments for the BioCreative VII Track 5 challenge (*Multi-label topic classification for COVID-19 literature annotation*) organized by the US NIH / National Library of Medicine [2]. The task, based on a sample from a manually curated database of COVID-19 studies [3], is to automatically assign one or more out of 8 topic labels (such as *Diagnosis*, *Treatment*, *Transmission*) based on an article's abstract, title and metadata.

II. Methods

We evaluated several methods based on machine learning, which included bag-of-words methods and the Transformer [11] based ones. The development set provided by the challenge organizers was used to compare the performance of the methods that we evaluated.

In addition to the provided training and development sets, we extended the training set recovering citations from LitCovid[1], which were already categorized according to the challenge categories. We ensured that our data set did not include any of the PMIDs of the test set. We used the following fields as the input for all models: *keywords*, *title*, and *abstract*.

A. MTI ML

Support Vector Machines (SVM) have been very popular in text categorization. We have evaluated a method that implements the training of a linear SVM using gradient descent and the modified Huber loss [12], [13]. Another characteristic is that features are binary, either a word appears in the document or not, which showed to be effective for MEDLINE citations [8]. The implementation of the SVM algorithm using gradient descent is based on the MTI ML package[2] provided by the US NIH/National Library of Medicine. This implementation is fast and can scale up to a large number of documents. We used our own revised version of MTI ML[3].

We combined unigrams and bigrams, removing stopwords and numbers, and we trained the system using the default parameters for 10 epochs. We trained and run MTI ML using commodity hardware. Results on the development set are available in table I. Here and in the tables below the best results across models are marked in bold.

TABLE I
MTI ML results per category on the development set.

Category	Precision	Recall	F1
Case Report	**0.9136**	0.8776	0.8952
Diagnosis	0.8489	0.8687	0.8587
Epidemic Forecasting	0.7949	0.6458	0.7126
Mechanism	0.8882	0.8444	0.8657
Prevention	0.9473	0.9222	0.9346
Transmission	0.6857	0.6563	0.6707
Treatment	0.8967	0.8854	0.8910

B. FastText

FastText [9] learns a representation from the input documents, which intends to improve on classifiers such as SVM but being faster than neural networks. This tool is fast and can be trained using commodity hardware.

In our experiments, we trained one model per category for 10 epochs. We configured it to combine unigrams and bigrams as features with a learning rate of 0.1. Results on the development set are available in table II.

C. SciBERT

We used SciBERT [1], which is a pre-trained BERT [5] system on biomedical literature. We used the pre-trained model *allenai/scibert_scivocab_uncased* available from HuggingFace[4] using the *BertForSequenceClassification* class. We trained one model per category for 30 epochs, using the

[1] https://www.ncbi.nlm.nih.gov/research/coronavirus

[2] https://lhncbc.nlm.nih.gov/ii/tools/MTI_ML.html

[3] https://github.com/READ-BioMed/MTIMLExtension

[4] https://huggingface.co

TABLE II
FASTTEXT RESULTS PER CATEGORY ON THE DEVELOPMENT SET.

Category	Precision	Recall	F1
Case Report	0.9127	0.8672	0.8894
Diagnosis	0.8562	0.8629	0.8595
Epidemic Forecasting	0.7905	0.6094	0.6882
Mechanism	0.8903	0.8397	0.8643
Prevention	0.9469	0.9204	0.9334
Transmission	0.6920	0.6055	0.6458
Treatment	0.8804	0.8872	0.8838

development set as reference to keep the trained model after each epoch. We used Adam with a learning rate of 2e-5 (which was decreased after each epoch). Results on the development set are available in table III. Result are better than the bag-of-words methods.

TABLE III
SCIBERT RESULTS PER CATEGORY ON THE DEVELOPMENT SET.

Category	Precision	Recall	F1
Case Report	0.8982	**0.9149**	**0.9065**
Diagnosis	**0.8745**	**0.8972**	**0.8857**
Epidemic Forecasting	0.8084	**0.7031**	**0.7521**
Mechanism	**0.9019**	**0.8826**	**0.8921**
Prevention	0.9458	**0.9513**	**0.9485**
Transmission	0.6128	0.7852	**0.6884**
Treatment	**0.8986**	**0.9311**	**0.9146**

D. Spectre

Specter [4] is a representation model for scientific documents, which uses citations to incorporate inter-document relations into document embeddings. As Specter has shown promising results for the classification of MeSH headings, we use it here to capture the inter-class similarity of articles. We used the pre-trained model *allenai/specter* from HuggingFace[5] to embed the documents, and then used the resulting embeddings as an input layer for a 3-layer classification FFN which we trained for 10 epochs. We used the same optimization and learning rate parameters as for SciBERT. The results on the development set are presented in table IV.

TABLE IV
SPECTER RESULTS PER CATEGORY ON THE DEVELOPMENT SET.

Category	Precision	Recall	F1
Case Report	0.9079	0.8589	0.8827
Diagnosis	0.8308	0.8700	0.8499
Epidemic Forecasting	**0.8165**	0.6719	0.7371
Mechanism	0.8573	0.8621	0.8597
Prevention	0.9481	0.9164	0.9320
Transmission	**0.6947**	0.5156	0.5919
Treatment	0.8801	0.9116	0.8956

E. BioELECTRA

We tried ELECTRA [**?**], which pre-trains text encoders as discrimintors rather than generators. We used the Bio-ELECTRA [10] pre-trained model *kamalkraj/bioelectra-base-discriminator-pubmed* available from HuggingFace[6] using the same implementation and parameters as SciBERT. The results on the development set are available in table V.

TABLE V
BIOELECTRA RESULTS PER CATEGORY ON THE DEVELOPMENT SET.

Category	Precision	Recall	F1
Case Report	0.9057	0.8963	0.9009
Diagnosis	0.8224	0.8868	0.8530
Epidemic Forecasting	0.8079	0.6354	0.7114
Mechanism	0.8743	0.8686	0.8714
Prevention	**0.9486**	0.9255	0.9369
Transmission	0.5971	**0.8047**	0.6855
Treatment	0.8978	0.8913	0.8945

F. Ensemble of classifiers

Combination of classifiers for MEDLINE citations has been shown to be effective [7]. We propose two methods for combining the output of the classifiers. One of them is to average the score of the classifiers, we ensure that the values of the scores are between 0 and 1. We experimented with several combinations of classifiers and found out that the combination of SciBERT, MTI ML and Specter produces more stable results. We noticed that SciBERT tended to give high scores to well-represented categories such as *Treatment* while giving scores close to zero for weaker classes such *Transmission*, so its performance varied greatly depending on the composition of the development set. On the other hand, such classifier as Specter and MTI ML were more conservative, assigning more scores close to 0.5 even for underrepresented categories, so they were more robust across datasets and had a higher precision for difficult categories. Thus by averaging their scores we create a voting system, where a class is likely to be assigned if at least one of the classifiers gives it a very strong score, or at least two of the classifiers give scores well over 0.5 but not necessarily close to 1.

In some cases, we observed that the score of the classifiers are quite polarised and decided to use as well the maximum score proposed by the combined classifiers. In this case, we removed the results of Specter, as it was assigning too many winning scores for the documents in the under-represented categories (its recall suffered in terms of false positives). Thus we take the maximum value of the scores from SciBERT and MTI ML in order to combine the benefits of the strong performance of the former across well-represented classes and the relatively high precision of the former for under-represented categories.

[5] https://huggingface.co/allenai/specter

[6] https://huggingface.co

III. Results

We successfully submitted 4 runs, which combined SciBERT and the MTI ML outputs, which were selected based on the results on the development set shown in the previous section and the success in running them on the test set on time.

- **uom_scibert_final:** Predictions are based on SciBERT as explained above.
- **uom_mtiml:** Predictions are based on the MTI ML method explained above. The predictions have been processed using the sigmoid function to provide a score between 0 and 1.
- **uom_averaged:** predictions from the SciBERT and MTI ML runs have been averaged. We observed that SciBERT weights are quite polarised, so averaging might not provide the best performance.
- **uom_max:** for this run, the maximum score from SciBERT and MTI ML are selected. This might be a better strategy compared to doing the average of the scored.

Results on the test set are available in tables VI, VII and VIII. We also show the results for the baseline method (ML-Net) based on shallow embeddings [6]. The best precision, recall and F1 score for each metric are marked in bold.

TABLE VI
Results on label based micro average for the test set provided by the challenge organizers.

Submission	Precision	Recall	F1
uom_scibert_final	0.9210	0.8219	0.8686
uom_mtiml	0.9136	0.8595	0.8857
uom_averaged	**0.9242**	0.8332	0.8764
uom_max	0.8861	**0.9143**	**0.9000**
ml-net	0.8756	0.8142	0.8437

TABLE VII
Results on label based macro average for the test set provided by the challenge organizers.

Submission	Precision	Recall	F1
uom_scibert_final	0.8111	0.5664	0.6027
uom_mtiml	0.8820	0.8398	0.8571
uom_averaged	**0.9533**	0.6318	0.6983
uom_max	0.8641	**0.8764**	**0.8669**
ml-net	0.8364	0.7309	0.7655

TABLE VIII
Instance based results on the test set provided by the challenge organizers.

Submission	Precision	Recall	F1
uom_scibert_final	0.8715	0.8415	0.8562
uom_mtiml	0.8976	0.8850	0.8913
uom_averaged	0.8874	0.8551	0.8710
uom_max	**0.9058**	**0.9316**	**0.9185**
ml-net	0.8849	0.8514	0.8678

IV. Discussion

MTI ML shows a better performance in the three tables compared to SciBERT, while the best method overall is the ensemble based on the maximum value. This performance might be due to the better performance of SciBERT on the most common categories, while MTI ML seems to perform better on the more infrequent ones that may be have a different distribution compared to the training and development sets. Overall, the ensemble based on the maximum value seems to be an effective strategy for recall, with scores greatly above the baseline and placing the system in the top quartile of all submitted results. On the other hand, the ensemble based on voting (average) of three systems had a high precision, especially for under-represented and challenging categories, which led to its very strong result in terms of macro precision.

V. Acknowledgements

We acknowledge the funding support of the US Army International Pacific Centre.

This research was undertaken using the LIEF HPC-GPGPU Facility hosted at the University of Melbourne. This Facility was established with the assistance of LIEF Grant LE170100200.

References

[1] Iz Beltagy, Kyle Lo, and Arman Cohan. SciBERT: A pretrained language model for scientific text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3615–3620, 2019.

[2] Qingyu Chen, Alexis Allot, Robert Leaman, Rezarta Islamaj Doğan, and Zhiyong Lu. Overview of the BioCreative VII LitCovid Track: multi-label topic classification for COVID-19 literature annotation. In *Proceedings of the seventh BioCreative challenge evaluation workshop*, 2021.

[3] Qingyu Chen, Alexis Allot, and Zhiyong Lu. LitCovid: an open database of COVID-19 literature. *Nucleic acids research*, 49(D1):D1534–D1540, 2021.

[4] Arman Cohan, Sergey Feldman, Iz Beltagy, Doug Downey, and Daniel S. Weld. SPECTER: Document-level Representation Learning using Citation-informed Transformers. In *ACL*, 2020.

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT (1)*, pages 4171–4186, 2019.

[6] Jingcheng Du, Qingyu Chen, Yifan Peng, Yang Xiang, Cui Tao, and Zhiyong Lu. ML-Net: multi-label classification of biomedical texts with deep neural networks. *Journal of the American Medical Informatics Association*, 26(11):1279–1285, 2019.

[7] Antonio Jimeno Yepes, James G Mork, Dina Demner-Fushman, and Alan R Aronson. Comparison and combination of several MeSH indexing approaches. In *AMIA annual symposium proceedings*, volume 2013, page 709. American Medical Informatics Association, 2013.

[8] Antonio Jimeno Yepes, Bartłomiej Wilkowski, James G Mork, Elizabeth Van Lenten, Dina Demner Fushman, and Alan R Aronson. A bottom-up approach to MEDLINE indexing recommendations. In *AMIA Annual Symposium Proceedings*, volume 2011, page 1583. American Medical Informatics Association, 2011.

[9] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431, Valencia, Spain, April 2017. Association for Computational Linguistics.

[10] Kamal raj Kanakarajan, Bhuvana Kundumani, and Malaikannan Sankarasubbu. Bioelectra: Pretrained biomedical text encoder using discriminators. In *Proceedings of the 20th Workshop on Biomedical Language Processing*, pages 143–154, 2021.

[11] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 6000–6010, 2017.

[12] Lana Yeganova, Donald C Comeau, Won Kim, and W John Wilbur. Text mining techniques for leveraging positively labeled data. In *Proceedings of BioNLP 2011 Workshop*, pages 155–163, 2011.

[13] T. Zhang. Solving large scale linear prediction problems using stochastic gradient descent algorithms. In *Proceedings of the twenty-first international conference on Machine learning*, page 116. ACM, 2004.