New York University Abu Dhabi

Division of Science and Mathematics

Document Classification for COVID-19 Literature

Author
Muyan Jiang *(mj2259)*
Omar Hussein *(oah242)*
Runyao Fan *(rf1888)*

Mentor
Prof. Dr. Adam Meyers

May 25, 2021

Abstract

The COVID-19 pandemic has led to a deluge of research publications related to various aspects of the virus. From research into COVID-19 transmission mechanisms to the public health response of various countries, there is a lot of information that needs to be categorized for easier and more efficient access to resources. This paper explores various machine learning based document classification techniques to categorize COVID-19 related literature. We integrate a novel terminology dictionary with the machine learning models to study the impact on the effectiveness of various classification techniques. We report a slight boost to F1 scores as a result of our modifications.

1 Introduction

The COVID-19 pandemic has had wide-ranging impacts on our world. Researchers around the globe have devoted themselves to investigating different academic areas related to the pandemic, ranging from clinical practice to epidemiological analysis. Many publications have been made available for people to collaborate together and tackle the virus. *LitCovid*[1][2], a curated literature hub for tracking up-to-date scientific information about the Coronavirus, indicates that more than 115,000 papers have been published.

Nowadays, with the abundance of available publications on COVID-19, it is particularly important for researchers to find and gauge relevant literature as easily and efficiently as possible. Therefore, we seek to develop a suitable document classification system focusing on COVID-19 literature. Specifically, we investigate whether having a terminology dictionary improves the efficacy of traditional machine learning approaches to document classification.

2 Literature Review

There has been significant work done to apply different document classification techniques to biomedical data to help researchers identify relevant information to a high degree of accuracy. With the high rate at which COVID-19 related literature is published, medical scientists cannot keep up with new publications and have to use Named Entity Recognition (NER) and Named Entity Normalization (NEN) to identify relevant publications [3].

One of the most popular paradigms currently used in the application of NLP models to classify biomedical data is the 'pretrain-and-finetune' approach [4]. Indeed, researchers have shown some success in applying this model to COVID-19 data classification tasks [5]. Specifically, their research indicates that there are measurable benefits to having a dedicated biomedical vocabulary base for biomedical document classification [5].

An additional approach to document classification has been the application of aspect-based document similarity measures. This has enabled researchers to perform pairwise document classification to identify aspects on which papers are more similar in order to achieve more fine-grained classification [6].

Our literature reviewed indicates that BioBERT works best as a classifier for COVID literature from the LitCovid database we utilise (achieving an F1 score is 86.1), followed by pre-trained language models and traditional machine learning algorithms in descending order of effectiveness [5].

3 Dataset

3.1 LitCovid

LitCovid is a dataset with more than 115,000 COVID-19 related papers collected and manually classified according to the following labels: *General, Transmission Dynamics (Transmission), Treatment, Case Report, Epidemic Forecasting (Forecasting), Prevention, Mechanism and Diagnosis*. From this link we were able to download a dataset with 52,419 entries for the training file, 8,226 entries for testing and 6,582 entries for validation. Each entry wraps the information of one paper including ID, journal name, title of a paper, abstract, keywords, label, publication type, authors, date, doi, label type, etc. This information, specifically title, abstract and keywords, is used later for feature creation in our model.

3.2 NCBI Disease Corpus

We utilised the NCBI Disease Corpus when generating terminology dictionaries for each of our classification categories. From this link we were able to download the NCBI Disease Corpus. It is a fully annotated biomedical research resource which contains 793 PubMed abstracts.

4 Method

4.1 Data Pre-processing

The LitCovid dataset is available as CSV files with fields including pmid, journal, title, abstract, keywords, label, publication type, authors, date1, doi and date2. The label field corresponds to each entry's category (e.g *General, Forecasting, etc*). As we want to predict each document's category, we design our model to predict the label field for each paper. After inspection of the other fields, we identify the title, abstract, and keyword fields as three fields that provide useful features for model training in machine learning.

In the initial processing steps we convert the strings in the selected three fields into tokens and remove less relevant tokens such as those containing less than 3 characters and words found in our stop-word dictionary. We use lemmatization and stemming to convert the tokens into their base forms.

To convert the tokens into input for various machine learning models, we use TF-IDF as a feature extraction method.

For our training set, each entry's category labels are converted into a vector representation. In the vector representation, categories associated with a particular entry are labelled as 1 and categories that are not represented are labelled as 0. For example, if a particular entry is classified under General and Transmission, both of these label have a value of 1 and all other category labels have a value of 0.

4.2 Initial Training With Traditional Machine Learning Algorithms

Our aim is to have our model return 0 or more labels for each input. In other words, we carry out a multi-label text classification task. There are two general approaches to multi-label classification tasks, namely problem adaption and problem transformation. For our problem adaption approach, we look at classification models that can return 0 or more labels for inputs.

The Scikit-learn (sklearn) library provides such classifiers, including k-nearest neighbour, decision trees, and random forest. We use these three classifiers with our input data and achieve varied degrees of success.

Moreover, we look at several binary classifiers to label our dataset by combining the binary classification output they produce. Essentially, we look at classifiers that predict each category individually and combine the outputs of those binary algorithms to represent a multi-label classification for each entry. Using models offered by the sklearn library, we experiment with bagging, gradient boosting, naïve Bayes, label power set, linear SVC, as well as classifiers that utilize linear SVC such as binary relevance and one-vs-rest classifiers.

We experiment with these various traditional machine learning models to determine which model produces the most accurate result before any modifications. We adopt the best performing model for further modification to see if we can better its results through the addition of a terminology dictionary.

4.3 Terminology Extraction

Given the academic characteristic of our dataset, we expect there to be a lot of biomedical jargon and terminology. As some studies suggest that having a dedicated biomedical dictionary can aid with classification tasks for biomedical datasets [5], we build and apply a tailored dictionary containing COVID-19 related terminology to the model in order to modify and improve the classification algorithm.

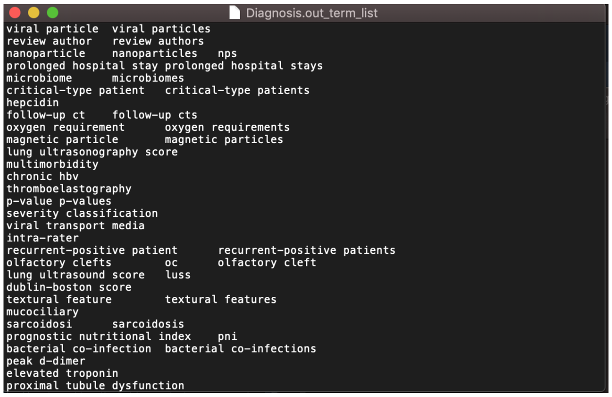
For terminology extraction, we utilize an open source tool for finding terminology in text called *The Termolator* [7] developed by Adam Meyers, Yifan He, Zachary Glass and Shasha Liao. The Termolator tool takes in two sets of documents. One set of documents acts as a foreground and the other acts as background. The Termolator extracts terminology that characterizes the foreground more than the background. As COVID-19 literature is a sub-field within the broader field of biomedical research data, it is important and helpful to maintain a subset-superset relationship between the foreground documents and background documents.

For our background documents we used 500 PubMed paper abstracts from *NCBI Disease Corpus* [8]. The NCBI Disease Corpus forms a superset of the foreground documents from the LitCovid dataset because the topic of the NCBI corpus is oriented more broadly towards medical papers while the foreground dataset focuses solely on COVID-19 data.

We use our LitCovid dataset as the source of our foreground documents and we construct a total of 9 terminology dictionaries with 10,000 words in each dictionary. Each dictionary corresponds to one of the 9 categories we use to label our data. Thus, each dictionary contains terminology that characterizes certain academic features of the corresponding category. For example, for the ”Diagonosis Dictionary”, we collect words such as ”follow-up CT”, ”prognostic nutritional index”, and ”thromboelastography” that are more closely related to diagnostic techniques. For another example, from ”Epidemic Forecasting Report Dictionary”, we collect words like ”fractional-order model”, ”disease-free equilibrium”, and ”controlled reproduction numbers”, which are more relevant to model prediction and theoretical forecasting. We provide a sample terminology dictionary for the diagnosis category below.

We use the same background documents from the NCBI Disease Corpus in combination with our 9 different foreground document sets to generate the 9 terminology dictionaries.

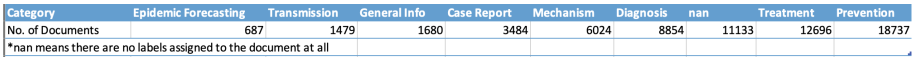
It is important to note that the number of documents in each category from the training set is not uniform which makes our training set unbalanced. For this reason, we choose



```
viral particle  viral particles
review author   review authors
nanoparticle    nanoparticles   nps
prolonged hospital stay prolonged hospital stays
microbiome      microbiomes
critical-type patient   critical-type patients
hepcidin
follow-up ct    follow-up cts
oxygen requirement      oxygen requirements
magnetic particle       magnetic particles
lung ultrasonography score
multimorbidity
chronic hbv
thromboelastography
p-value p-values
severity classification
viral transport media
intra-rater
recurrent-positive patient      recurrent-positive patients
olfactory clefts        oc      olfactory cleft
lung ultrasound score   luss
dublin-boston score
textural feature        textural features
mucociliary
sarcoidosi      sarcoidosis
prognostic nutritional index    pni
bacterial co-infection  bacterial co-infections
peak d-dimer
elevated troponin
proximal tubule dysfunction
```

Figure 1: Diagnosis Terminology Dictionary

to use all documents from each category in the LitCovid dataset to act as the foreground for the corresponding terminology dictionary, instead of a random sample. The following table displays the distribution of the documents in the training set by category. Note that since one document can have multiple labels, the total here exceeds the total number of documents.



Category	Epidemic Forecasting	Transmission	General Info	Case Report	Mechanism	Diagnosis	nan	Treatment	Prevention
No. of Documents	687	1479	1680	3484	6024	8854	11133	12696	18737
*nan means there are no labels assigned to the document at all									

Figure 2: Documents Distribution w.r.t. Categories

4.4 NLP Modification/Correction with Terminology Dictionaries in SVC

After determining the best performing traditional machine learning classification model (i.e. SVC Sq. Hinge Loss), we use the above-mentioned 9 terminology dictionaries to implement our own novel modification.

Given a new input (i.e. an abstract from the test set), we look at all 9 dictionaries and generate a TF-IDF based similarity vector that indicates the relevance of the input to the dictionaries. We standardize the vectors and incorporate the results into the SVC classification process as follows.

We access the decision function of the SVC, which is a vector of 9 entries that indicates the distance of the input to be classified to the decision boundaries (9 One-vs-Rest hyperplanes in the context of multi-class classification) and add the standardized similarity function by a factor of α. Namely, the corrected decision vector becomes ”original decision vector + α * standardized similarity vector”. This α is a hyperparameter to be tuned and it helps us avoid influencing the initial decision function too much. We also experiment with tanh (similarity vector) to sequence the entries between -1 and 1. The effect of this is discussed later.

As a result, when an input is very close to the decision boundary but situates incorrectly on the other side, we correct it by adding this standardized similarity vector. In theory, if one document is very ”close” to, say, dictionary A, then the similarity at the corresponding bit in the standardized similarity vector is supposed to be very positive. If this document is very "far" from dictionary B, then the corresponding bit is supposed to be very negative. These numerical representations have corrective effects on the decision function and thus enhance the performance of the new model.

The overall process is illustrated by the flow chart in Figure 3 below.

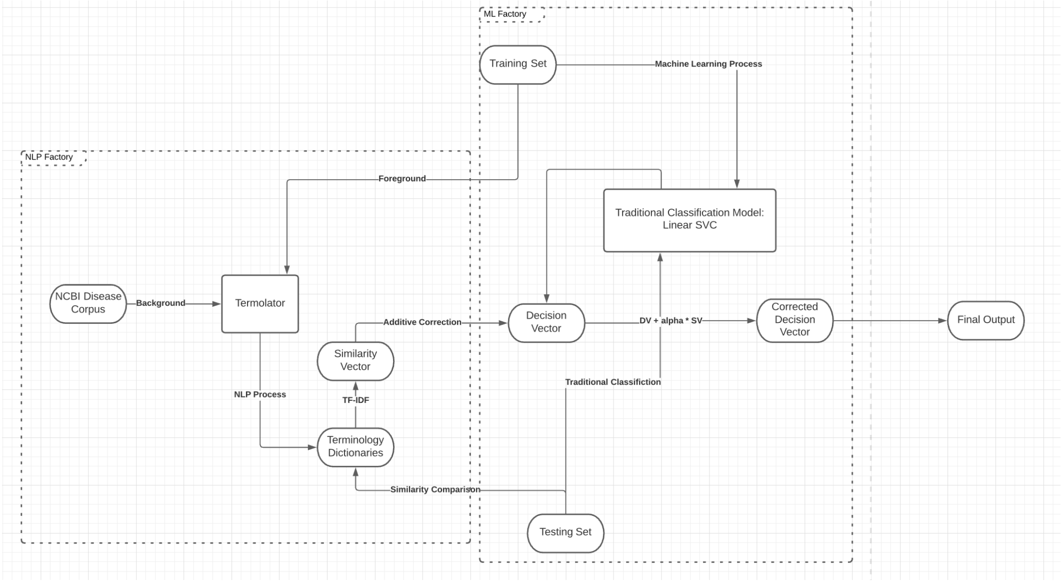


Figure 3: Modified Algorithm Diagram

5 Result and Evaluation

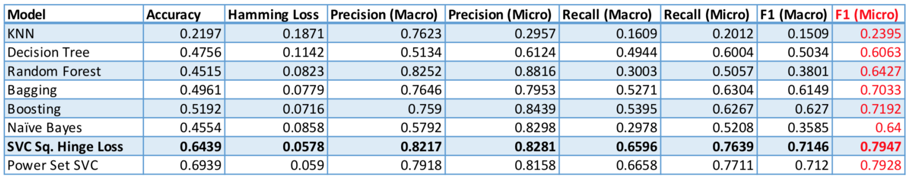
5.1 Model Evaluation

In a multi-label/multi-class classification setting, *Micro-Averaging* and *Macro-Averaging* are typically used as the performance evaluation metrics. A macro-average will compute the metric independently for each class and then take the average (hence treating all classes equally). However, a micro-average will aggregate the contributions of all classes to compute the average metric. In our situation, we mainly focus on micro-averaging due to the unbalanced number of documents in each class (e.g, "Treatment" class has over 10,000 documents while "Epidemic Forecasting" only consists of approximately 700 documents).

Additionally, we use another indication flag – *Hamming-Loss*. In multi-class classification, Hamming Loss corresponds to the Hamming distance between the ground truth and the predicted values. Namely, it is the fraction of labels that are incorrectly predicted (i.e. the fraction of wrong labels to the total number of labels).

5.2 Model Comparison and Results

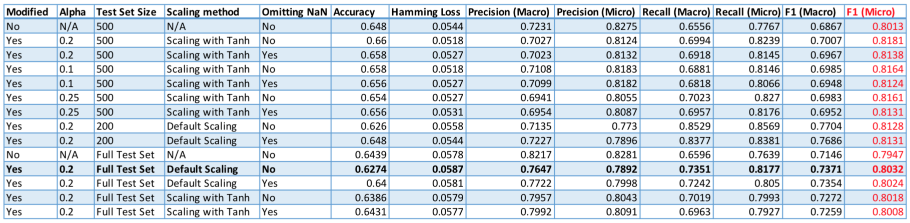
From our initial evaluation of unmodified traditional machine learning algorithms, we see that SVC Sq. Hinge Loss and Power Set SVC models give good overall results compared to the other models. Looking at the F1 (Micro) score, we see that all models except the k-nearest-neighbour classifier gives a score of higher than 0.6. Nevertheless, similar to the test results we obtained through preliminary training using only the titles as features, the best-performing models when we consider the comprehensive scoring are SVC Sq. Hinge Loss and Power Set SVC. Models like Random Forest and Naive Bayes give relatively poor performance and this could be linked to the high correlation between the labels.



Model	Accuracy	Hamming Loss	Precision (Macro)	Precision (Micro)	Recall (Macro)	Recall (Micro)	F1 (Macro)	F1 (Micro)
KNN	0.2197	0.1871	0.7623	0.2957	0.1609	0.2012	0.1509	0.2395
Decision Tree	0.4756	0.1142	0.5134	0.6124	0.4944	0.6004	0.5034	0.6063
Random Forest	0.4515	0.0823	0.8252	0.8816	0.3003	0.5057	0.3801	0.6427
Bagging	0.4961	0.0779	0.7646	0.7953	0.5271	0.6304	0.6149	0.7033
Boosting	0.5192	0.0716	0.759	0.8439	0.5395	0.6267	0.627	0.7192
Naïve Bayes	0.4554	0.0858	0.5792	0.8298	0.2978	0.5208	0.3585	0.64
SVC Sq. Hinge Loss	**0.6439**	**0.0578**	**0.8217**	**0.8281**	**0.6596**	**0.7639**	**0.7146**	**0.7947**
Power Set SVC	0.6939	0.059	0.7918	0.8158	0.6658	0.7711	0.712	0.7928

Figure 4: Results of Different Models

We identify SVC Sq. Hinge Loss as the best-performing traditional machine learning algorithm so we apply our terminology dictionary based modification on the SVC Sq. Hinge Loss model. In generating the modified vector, the parameters related to the correction process include the relative weight of the SVC model's prediction; our dictionary similarity vector; the method we use to scale the dictionary similarity vector, as well as whether to ignore the value of the NaN entry in the dictionary similarity vector. By changing these parameters, we obtain performance scores for a variety of parameter settings.



Modified	Alpha	Test Set Size	Scaling method	Omitting NaN	Accuracy	Hamming Loss	Precision (Macro)	Precision (Micro)	Recall (Macro)	Recall (Micro)	F1 (Macro)	F1 (Micro)
No	N/A	500	N/A	No	0.648	0.0544	0.7231	0.8275	0.6556	0.7767	0.6867	0.8013
Yes	0.2	500	Scaling with Tanh	No	0.66	0.0518	0.7027	0.8124	0.6994	0.8239	0.7007	0.8181
Yes	0.2	500	Scaling with Tanh	Yes	0.658	0.0527	0.7023	0.8132	0.6918	0.8145	0.6967	0.8138
Yes	0.1	500	Scaling with Tanh	No	0.658	0.0518	0.7108	0.8183	0.6881	0.8146	0.6985	0.8164
Yes	0.1	500	Scaling with Tanh	Yes	0.656	0.0527	0.7099	0.8182	0.6818	0.8066	0.6948	0.8124
Yes	0.25	500	Scaling with Tanh	No	0.654	0.0527	0.6941	0.8055	0.7023	0.827	0.6983	0.8161
Yes	0.25	500	Scaling with Tanh	Yes	0.656	0.0531	0.6954	0.8087	0.6957	0.8176	0.6952	0.8131
Yes	0.2	200	Default Scaling	No	0.626	0.0558	0.7135	0.773	0.8529	0.8569	0.7704	0.8128
Yes	0.2	200	Default Scaling	Yes	0.648	0.0544	0.7227	0.7896	0.8377	0.8381	0.7686	0.8131
No	N/A	Full Test Set	N/A	No	0.6439	0.0578	0.8217	0.8281	0.6596	0.7639	0.7146	0.7947
Yes	**0.2**	**Full Test Set**	**Default Scaling**	**No**	**0.6274**	**0.0587**	**0.7647**	**0.7892**	**0.7351**	**0.8177**	**0.7371**	**0.8032**
Yes	0.2	Full Test Set	Default Scaling	Yes	0.64	0.0581	0.7722	0.7998	0.7242	0.805	0.7354	0.8024
Yes	0.2	Full Test Set	Scaling with Tanh	No	0.6386	0.0579	0.7957	0.8043	0.7019	0.7993	0.7272	0.8018
Yes	0.2	Full Test Set	Scaling with Tanh	Yes	0.6431	0.0577	0.7992	0.8091	0.6963	0.7927	0.7259	0.8008

Figure 5: Results of Different Parameter Combinations

In order to determine an appropriate value of alpha, which determines the relative weight between the model-generated decisions and the dictionary similarity vector, we use multiple values of alpha on a test set of size 500 and obtain varying performances. Some values of alpha, such as 0.5 and 1, obtain significantly worse performance and are omitted in the graph. We observe that for the three alpha values 0.1, 0.2, and 0.25, performance for models with alpha = 0.2 is comparable to the other two in all individual metrics and is slightly better in micro F1 score. As such, we set alpha as 0.2.

We run the model on the entire test set with varying scaling methods as well as ways to handle the NaN category in the dictionary similarity vector. The default scaling provided by preprocessing.scale() function may generate values larger than 1 or smaller than -1 so we experiment using the tanh function to transform the scaled vectors such that all values are kept between -1 and 1. Moreover, we experiment with ignoring the NaN category in the dictionary similarity vector because preliminary observations show that the value of NaN is often disproportionately large in the generated dictionary similarity vectors.

We observe that the models with terminology dictionary correction obtained better micro

F1 score on the entire test set compared to the original model. With regards to the NaN entry, models that ignore these entries perform better in accuracy, hamming loss and precision but worse in recall and F1 score. Similarly, the models using tanh function perform better in some measurements but have slightly worse performance in terms of F1 scores.

A model may have better performance in certain aspects but worse performance in others when compared to another model because the measurements consider different things. For example, precision and recall need an output vector to be exactly the same as the actual vector to be counted as right, while hamming loss looks at how similar output vectors are to actual vectors by comparing bit by bit. Micro F1 score is relatively more important in our evaluation process and by this measurement, our new model that incorporates terminology dictionary correction performs better than the original SVC Sq. Hinge Loss model.

To further study the performance of our current model and explore possibilities for future improvement, we generate confusion matrices for each of the nine labels. Each confusion matrix shows the performance of our model with respect to a particular label. For example, the confusion matrix of label ”prevention” shows the number of true positive, false positive, true negative, and false negative cases generated by our model with respect to ”prevention”.

We notice that while most negatives can be correctly classified as negatives, the model performs less ideally when predicting positives. For example, 25 out of 36 documents with label ”General Info” are predicted as not having the label ”General Info”. This could be because documents classified as general info do not have many characteristic features and vocabulary, and the solutions to these classification errors can be a direction for our future improvement.

Furthermore, as noted previously, our training set is heavily unbalanced. Thus, categories with larger datasets (e.g. Prevention) perform much better than categories with smaller datasets (e.g. General Info).

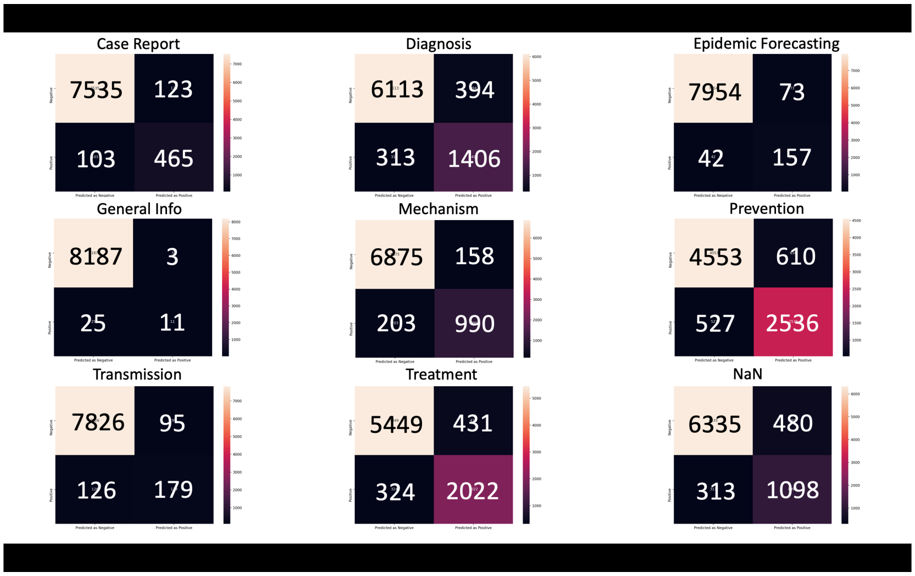


Figure 6: Confusion Matrices of Nine Labels (Top Left: True Negative; Top Right: False Positive; Bottom Left: False Negative; Bottom Right: True Positive).

In terms of comparing our results to the state-of-the-art, we managed to achieve an F1 score of 80.32 compared with BioBERT's 86.2 [5]. Considering the significant sophistication of BioBERT, especially compared to our much simpler model, our model provides reasonable results.

6 Future Work

There are certain limitations and potential extensions of our model. When utilizing the terminology dictionaries, we have not explored their full capacity. For example, The Termolator is capable of ranking the characteristics of terminology which means the words that appear earlier in the dictionary are more representative of the corresponding category. This information can be used to assign weighted relevance to terminology so for different documents that are compared against the dictionary, we could generate more targeted and reliable similarity vectors. In our current approach, we blindly use the dictionaries as a whole set and generate TF-IDF based similarity vectors.

Another potential extension with terminology dictionaries would be a different way of incorporating the similarity vector. For now, we only experiment with two ways: weighted addition of the similarity vector and tanh addition of the similarity vector. There might be a better strategy to experiment with correction method.

7 Conclusion

From the result of different models, it is clear to see that we have improved the SVC Sq. Hinge Loss model by generating terminology dictionaries and combining the system predictions with the dictionary similarity vectors we generate for each data point in the test set. Considering the micro F1 score, which reflects the performance of a multi-label classification task, we boost the F1 score to 0.8032 which is an improvement from the already sophisticated classification model.

References

[1] Q. Chen, A. Allot, and Z. Lu, "Litcovid: An open database of covid-19 literature," *Nucleic Acids Research*, 2020.

[2] Q. Chen, A. Allot, and Z. Lu, "Keep up with the latest coronavirus research," *Nature*, vol. 579, no. 7798, p. 193, 2020, ISSN: 1476-4687 (Electronic) 0028-0836 (Linking). DOI: 10.1038/d41586-020-00694-1. [Online]. Available: https://www.ncbi.nlm.nih.gov/pubmed/32157233.

[3] N. Colic, L. Furrer, and F. Rinaldi, "Annotating the pandemic: Named entity recognition and normalisation in COVID-19 literature," in *Proceedings of the 1st Workshop on NLP for COVID-19 (Part 2) at EMNLP 2020*, Online: Association for Computational Linguistics, Dec. 2020. DOI: 10.18653/v1/2020.nlpcovid19-2.27. [Online]. Available: https://www.aclweb.org/anthology/2020.nlpcovid19-2.27.

[4] P. Lewis, M. Ott, J. Du, and V. Stoyanov, "Pretrained language models for biomedical and clinical tasks: Understanding and extending the state-of-the-art," in *Proceedings of the 3rd Clinical Natural Language Processing Workshop*, Online: Association for Computational Linguistics, Nov. 2020, pp. 146–157. DOI: `10.18653/v1/2020.clinicalnlp-1.17`. [Online]. Available: `https://www.aclweb.org/anthology/2020.clinicalnlp-1.17`.

[5] B. Jimenez Gutierrez, J. Zeng, D. Zhang, P. Zhang, and Y. Su, "Document classification for COVID-19 literature," in *Findings of the Association for Computational Linguistics: EMNLP 2020*, Online: Association for Computational Linguistics, Nov. 2020, pp. 3715–3722. DOI: `10.18653/v1/2020.findings-emnlp.332`. [Online]. Available: `https://www.aclweb.org/anthology/2020.findings-emnlp.332`.

[6] M. Ostendorff, T. Ruas, T. Blume, B. Gipp, and G. Rehm, "Aspect-based document similarity for research papers," in *Proceedings of the 28th International Conference on Computational Linguistics*, Barcelona, Spain (Online): International Committee on Computational Linguistics, Dec. 2020, pp. 6194–6206. DOI: `10.18653/v1/2020.coling-main.545`. [Online]. Available: `https://www.aclweb.org/anthology/2020.coling-main.545`.

[7] P. Mayr, C. Zhang, A. Meyers, Y. He, Z. Glass, J. Ortega, S. Liao, A. Grieve-Smith, R. Grishman, and O. Babko-Malaya, "The termolator: Terminology recognition based on chunking, statistical and search-based scores," *Frontiers in Research Metrics and Analytics*, vol. 3, Jun. 2018. DOI: `10.3389/frma.2018.00019`.

[8] R. I. Doğan, R. Leaman, and Z. Lu, "Special report: Ncbi disease corpus: A resource for disease name recognition and concept normalization," *J. of Biomedical Informatics*, vol. 47, pp. 1–10, Feb. 2014, ISSN: 1532-0464.