SINAI at BioCreative VII LitCovid Track: Corpus augmentation for COVID-19 literature multi-label classification

Mariia Chizhikova, Pilar López-Úbeda, Manuel Carlos Díaz-Galiano, L. Alfonso Ureña-López and M. Teresa Martín-Valdivia

SINAI Group, Department of Computer Science, Advanced Studies Center in ICT (CEATIC)
Universidad de Jaén, Campus Las Lagunillas s/n, E-23071, Jaén, Spain

***Abstract*— *The rapid growth of the amounts of COVID-19 related literature has not only increased the burden of manual topic annotation, but even reached the point in which the need of an automatic annotation system has become evident. Leveraging knowledge from biomedical publications is an important step toward promoting the investigation and providing a better and more efficient research experience. This paper describes participation of the SINAI team in the Track 5, LitCovid Track, of the BioCreative VII competition. The challenge brought our effort to the task of multi-label topic classification for COVID-19 literature annotation. Our solution is based on a problem-transformation method that considers the prediction of each label as an independent binary classification task. This approach allowed us to use the Logistic Regression algorithm based on Term Frequency - Inverse Document Frequency (TF-IDF) representation of the tokenized and stemmed text data which was previously subjected to a corpus augmentation process. The almost inappreciable amount of time and computational resources our classifier takes to be trained gives a response to the high-speed LitCovid growth, and its performance (0.91 label-based micro average precision) will be an improvement beneficial to curators and researchers.*

Keywords*— *logistic regression, corpus augmentation, multi-label classification, back translation.

I. Introduction

A timely access to the rapidly growing amount of the scientific literature in the context of the ongoing COVID-19 pandemic contributes to enhancing the research. LitCovid (1), an open-resource coronavirus related literature hub, was created to address the need of a rapid and efficient management and consultation of such a vast amount of information. The fact that the clinical society needs new tools to keep aware of the latest research results in the field sets a challenge for Biomedical Natural Language Processing (BioNLP). Text classification is one of the common tasks in BioNLP and the multi-label classification has attracted more attention in recent time due to the complexity of the text semantics: the versatility that in varying degrees characterises every scientific investigation makes nearly impossible that one article would cover only one topic.

Track 5 of the BioCreative VII competition brought our effort to designing a system for automated topic annotation of the LitCovid articles. The goal was to label each of the articles with one or more labels and to achieve that our team opted for a problem transformation method that considers the prediction of each label as a binary classification task.

This paper describes the system presented by the SINAI team for the LitCovid track. Our solution is based on the Logistic Regression algorithm which takes the Term Frequency - Inverse Document Frequency (TF-IDF) representation of tokenized and stemmed text of the PubMed articles' abstracts. To improve the performance of our Machine Learning (ML) algorithm we proceeded with a corpus augmentation heuristics based on back translation and synonym replacements.

The structure of this article obeys its main objective: in Section II we will cover the pipeline designed to confront the given classification task and in Section III we will present the system performance which will lead us to the result analysis provided in Section IV.

II. System Overview

A. Dataset

The participants of the LitCovid track Multi-label topic classification for COVID-19 literature annotation were provided with training and development datasets containing 24,960 and 6,239 articles from LitCovid, respectively (2). The data was provided in CSV format with the following fields:

- pmid: PubMed Identifier
- journal: journal name
- title: article title
- abstract: article abstract
- keywords: author-provided keywords
- pub_type: article type
- authors: author names
- doi: Digital Object Identifier
- label: annotated topics

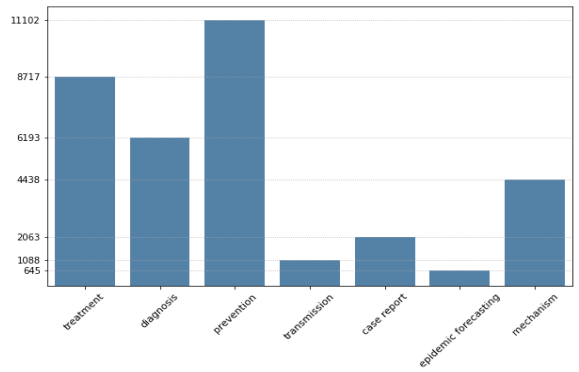

Fig. 1. Distribution of articles per classes in the training dataset.

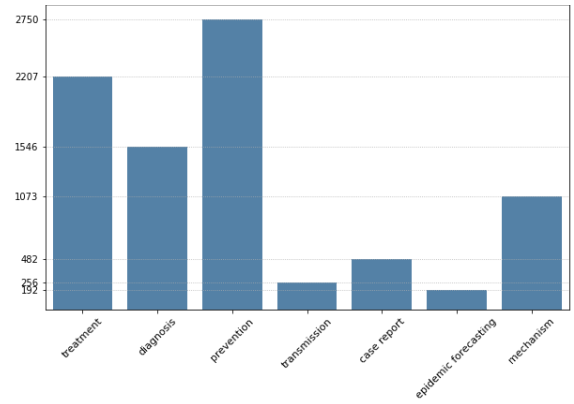

Fig. 2. Distribution of articles per classes in the development dataset

Each article was labeled with one or more of the subsequent topics separated by a semicolon: Treatment, Diagnosis, Prevention, Mechanism, Transmission, Epidemic Forecasting, and CaseReport. The distribution of articles per labels in training and development datasets can be seen on Figures 1 and 2 respectively.

As for the test set, it consisted of 2,500 entries with the same fields as the training and development ones, except annotated topics.

It is worth mentioning that for the submission run we used a concatenation of the training and development data as training set, while all the experiments were carried out using the development dataset as a test one in order to evaluate the resulting model.

B. *Pre-processing*

The initial step of every NLP solution is data preprocessing. In our particular case, we will be working with abstracts provided in the datasets, as they are a short but consistent overview of the topics covered in each article. The pre-processing applied to all texts is the following:

- *Lowercasing*. All texts were converted to lowercase.
- *Word tokenization*. In this step we split the strings in token objects using the RegexpTokenizer from the Natural Language Toolkit (NLTK) Python library. The separator was chosen to discard punctuation signs and special characters in order to skip the normalization step.
- *Stopwords removal.* Stopwords are the most commonly occuring words without lexical meaning which in our case introduce unnecessary noise and therefore should be removed. For this step, we used the list of stopwords provided by the NLTK[1] enriched with a custom list of the different names of the coronavirus disease, such as “covid-19” and “covid” that will be frequently mentioned in our corpus but are unlikely to shed light on articles’ topics.
- *Stemming*. This technique allows us to normalize the text, reducing the full form of a word to its stem by stripping the root of its derivational and inflectional affixes. It also is a crucial step for the further TF-IDF representation, since it addresses the sparsity issue. To normalize the vocabulary of each text we have used the implementation of the Porter Stemmer algorithm provided by the NLTK library. (7)

C. *Feature extraction and weighting. TF-IDF*

TF-IDF is used to calculate the weight of features that categorize a text in a collection. The more a stem appears in a pre-processed text, the more it is estimated to be significant in it. At the same time, the terms that are commonly used in a collection of texts are less relevant for the classification task (9).

In our system, we used the TfidfVectorizer provided by Scikit-learn[2]. The vectorizer was fitted using the full collection (abstracts from training, development and test dataset) for more precise IDF estimation.

D. *Logistic regression classifier*

Logistic Regression (LR) is a method to predict dichotomous result variables which in our case stand for the labels of each text (5). We trained a LR classifier with the Stochastic Average Gradient (SAG) solver - a randomized variant of the incremental aggregated gradient method (10). The regularization technique chosen is Ridge Regression and the inverse regularization strength parameter was set to 1 to get a classifier with smaller values of weights and better generalization ability.

During the development process, the classifier was fitted with data provided in the training set and tested on the development set. Despite the apparent simplicity of the LR classifier, we achieved encouraging results in labelling the most commonly present topics of our dataset, namely “treatment”, “diagnosis”, “prevention” and “mechanism” which can be observed on Figure 3.

[1] http://nltk.org

[2] http://scikit-learn.org

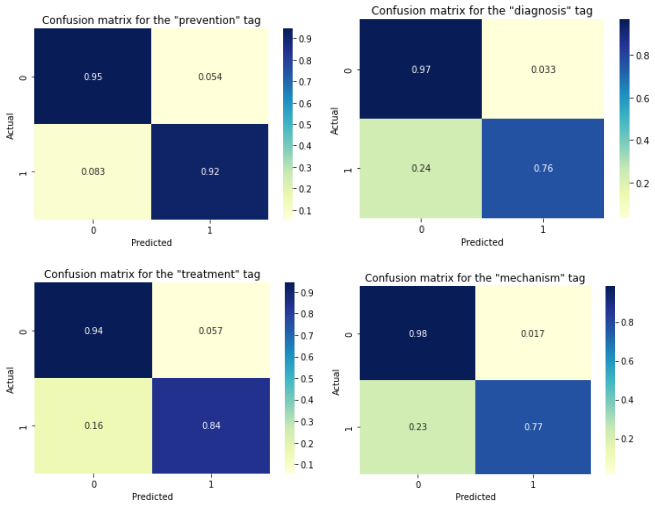

Fig. 3 Confusion matrices for the most commonly observed topics.

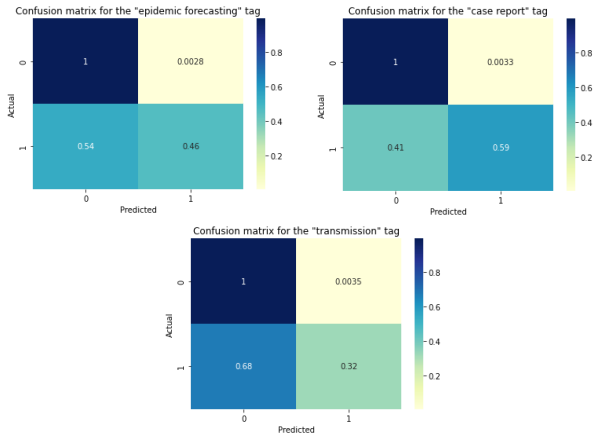

Fig. 4 Confusion matrices for the less commonly observed topics.

However, the performance of the model on the other classes resulted poorer, as can be seen on Figure 4. The frequency of appearance of each of three topics in question among the corpus allows us to typify these as 'rare events' which lead to a prediction problem. The following subsection provides a review of the methodology used to tackle rare event detection.

E. Corpus Augmentation

As can be inferred from Figures 1 and 2, the amounts of articles tagged by each label are far from being similar: while the most common label "prevention" was assigned to 11,102 articles from the training set, the less common topic "epidemic forecasting" is raised in only 645 corpus entries.

The class imbalance problem is known to hinder the performance of classification algorithms, since the modern ML techniques focus on minimizing the error rate of the majority class while ignoring the minority class (8). This phenomenon was indeed the cause of a large number of False Negatives when it came to "transmission", "case report" and "epidemic forecasting" topics.

To improve rare event detection we considered making use of two data augmentation techniques: Back Translation (BT) and synonym replacement (SR), both aimed to create new texts with the same meaning to preserve the original labelling. Data augmentation was performed on an extract from the training and development datasets containing 5,719 (3,796 from the training set and 1,923 from the development set) entries tagged with 3 less common labels mentioned above.

BT is a corpus augmentation method aimed to generate more text data by translating a given collection from the source language to another one and back. In our case, 5,719 English texts were translated to Spanish and back using the newest multilingual MarianMt models provided by Hugging Face[3] (13). Marian is a Neural Machine Translation framework written entirely in C++ with an integrated automatic differentiation engine based on dynamic computation graphs (6). We used 2 models able to translate text from English to any romance language (as was mentioned above, we chose Spanish as target language) and vice versa. Once this step was completed, the new abstracts were subjected to the same pre-processing procedure used on the original data for further elimination of duplicates. After this, we got 3,796 additional entries to our training set, which was still insufficient to balance it.

The following step in the data augmentation process consisted of replacement of every noun in each text with a synonym found in WordNet - a database that links English lexical words to sets of synonyms (synsets) that are turn linked through semantic relations that determine word definitions (12). For this purpose, each abstract was tokenized and labeled with its part-of-speech (POS) tag using the tagger provided by the above cited library. As no lexical word in any language has a single meaning, a disambiguation algorithm is needed to find an appropriate synset for each noun. To address the ambiguity, every noun was analysed by the Lesk algorithm which returns a synset with a sense definition that is similar to other words in the sentence that contains the noun in question (11). That synset was used to extract a synonym to replace the original noun. Word replacement based on its semantic and paradigmatic relation within the context enriches the text corpus with new terms, semantically relevant to the classification task.

As a result of applying both augmentation methods, we generated 6,134 new corpus entries which were concatenated with the original dataset. The distribution of articles per classes in this upgraded train set (31,094 entries in total) can be seen in Figure 5. It is worth mentioning that the process of dropping the duplicates affected the number abstracts corresponding to all labels and is the reason for a slight decrease of quantity members of the "treatment" category, for instance.

[3] https://huggingface.co/

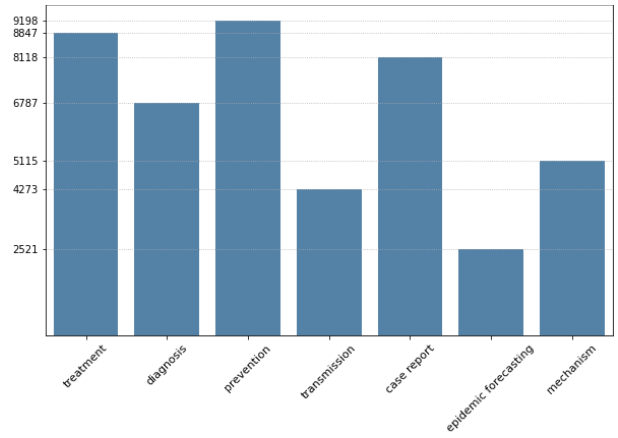

Fig. 5. Distribution of articles per classes in the updated dataset.

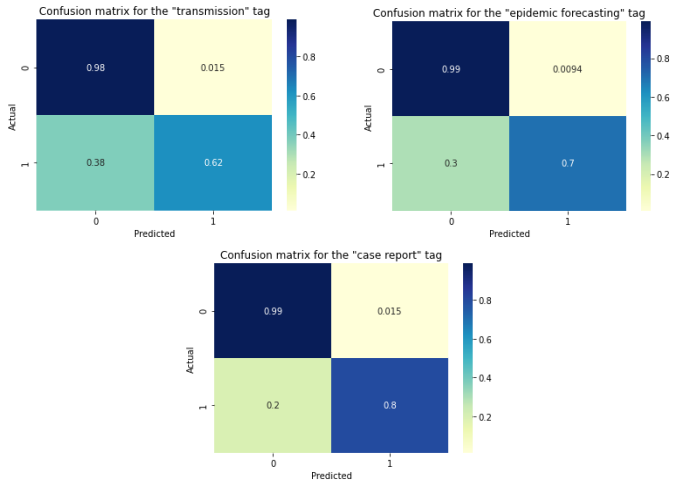

Fig. 6. Confusion matrices illustrating improved rare event detection

Label-based micro avg.	**Prescision.**	**Recall**	**F1**
Run 1	0.9111	0.7486	0.8219
Run 2	0.91	0.7553	0.8254
Run 3	0.91	0.7547	0.8251
Run 4	0.8991	0.737	0.81
ML-NET	0.8756	0.8142	0.8437
Label-based macro avg.	**Prescision.**	**Recall**	**F1**
Run 1	0.9004	0.6279	0.7203
Run 2	0.884	0.684	0.7603
Run 3	0.8829	0.6887	0.7643
Run 4	0.8343	0.7227	0.7629
ML-NET	0.8364	0.7309	0.7655
Instance-based	**Prescision.**	**Recall**	**F1**
Run 1	0.8334	0.782	0.8069
Run 2	0.8296	0.7887	0.8086
Run 3	0.829	0.7891	0.8086
Run 4	0.8006	0.7676	0.7838
ML-NET	0.8849	0.8514	0.8678

Fig. 7. System performance results.

III. Results

A. The impact of corpus augmentation

Even though the distribution of abstract per topics in the final training set was still not completely homogeneous, the experiments run with this data showed the improvements regarding the detection of the three labels we were working on, as can be seen in Figure 6.

B. Submission results

As we have already mentioned, for the submission run we added to the training set all the entries of the development set, which was also subjected to the data augmentation process described above. The final dataset we used to train our classifier contained 33,555 entries.

The metrics defined by the challenge to evaluate the submitted experiments are those commonly used for some NLP tasks such as text classification, namely precision, recall, and F1-score (F1) considering micro average and macro average. Moreover, instance-based precision, recall and F1-score were used to calculate average distance between true labels and predicted labels of each, averaged over all the training instances.

Furthermore, evaluation was carried out from the predicted probabilities of each tag and not from the binary result of their presence or absence. Figure 8 summarises the results of our submission compared with the performance of ML-NET - a deep learning framework for multi-label classification of biomedical texts selected as a baseline of the competition (4).

For the text classification task we submitted 4 LR classifier trained on different dataset:

- Run 1: LR classifier trained on the concatenation of both original training and development datasets.
- Run 2: LR classifier trained on SR augmentation dataset[4] extended with development data subjected to both augmentation techniques.
- Run 3: LR classifier trained on BT augmentation dataset extended with development data subjected to both augmentation techniques.
- Run 4: LR classifier trained on the concatenation of training and development data subjected to both augmentation techniques.

Figure 7 summarises the evaluation of our system's performance. The extension of corpus led to a slight improvement of label-based macro avg. recall when evaluated in test with the dataset provided by the organization committee. Nevertheless, there are no considerable improvements if we compare other values. The reason for this may be a small number of test set entries corresponding to the three categories we were working on, so the correct recognition of them has less impact on the average metrics. This assumption cannot be proven based on the information we have about the test dataset.

IV. Conclusions

This paper presents the systems proposed by the SINAI team to tackle multi-label topic annotation of articles from

[4] The term *augmentation dataset* refers to the training set concatenated with its extract that was subjected to corpus augmentation methods described above.

LitCovid corpus. The almost inappreciable amount of time and computational resources our classifier takes to be trained gives a response to the high-speed LitCovid growth. This fact can be considered an advantage over more time and resource-consuming Deep Learning frameworks. Overall, the described systems are able to accurately annotate topics of LitCovid articles analyzing their abstracts.

References

1. Bird, S., Loper, E., Klein, E. (2009), Natural Language Processing with Python. O'Reilly Media Inc. pp.D1534-D1540.2. Chen Q., Allot A., & Lu Z. Keep up with the latest coronavirus research. Nature. 2020 Mar; 579(7798): 193-193
3. Chen, Q., Allot, A. and Lu, Z., 2021. LitCovid: an open database of COVID-19 literature. Nucleic acids research, 49(D1)
4. Du, J. et. al., 2019. ML-Net: multi-label classification of biomedical texts with deep neural networks. Journal of the American Medical Informatics Association, 26(11), pp.1279-1285.
5. Hilbe, J. M. (2009). Logistic regression models. Chapman and hall/CRC.
6. Junczys-Dowmunt, M., Grundkiewicz, R., Dwojak, T., Hoang, H., Heafield, K., Neckermann, T., & Birch, A. (2018). Marian: Fast neural machine translation in C++. arXiv preprint arXiv:1804.00344.
7. Porter, M.F. (1980), "An algorithm for suffix stripping", Program, Vol. 14 No.3, pp. 130-137.
8. Thabtah, F. et al. Data imbalance in classification: Experimental evalu-ation. Information Sciences. 2020 Nov; 513: 429–441.
9. Salton, G., & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. Information processing & management, 24(5), 513-523.
10. Schmidt, M., Le Roux, N., & Bach, F. (2017). Minimizing finite sums with the stochastic average gradient. Mathematical Programming, 162(1-2), 83-112.
11. Lesk, M. "Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone." Proceedings of the 5th Annual International Conference on Systems Documentation. ACM, 1986
12. Miller, G. A. (1995). WordNet: a lexical database for English. Communications of the ACM, 38(11)
13. Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., ... & Rush, A. M. (2019). Huggingface's transformers: State-of-the-art natural language processing. arXiv preprint arXiv:1910.03771.