Classifying the stance and argument toward COVID-related fields using NLP methods

Karzanov Daniil
dvkarzanov@gmail.com

Abstract

This paper describes several approaches to solving the multi-variable classification problem within the RuArg22 competition organized by the Skolkovo Institute of Science and Technology. As a part of the solution, different pre-trained embeddings from HuggingFace were applied and compared. Various statistical learning models such as SVM, LOGIT, FastText, as well as neural network architecture, with class-balancing were exploited and compared in terms of prediction performance.

Keywords: nlp, argumentation mining, neural networks, hugging face

1 Introduction

The topic of covid, quarantine and masks seems to be still very relevant today despite the decreasing trend in the epidemics. Being able to classify people's positions in an automatic manner can still benefit different healthcare institutions in developing new strategies for promoting defensive measures against COVID.

As part of the RuArg22 competition, we are given data consisting of Russian texts posted in different media and addressing three different topics related to COVID-19 pandemics: vaccination, quarantine and masks. For each topic, we are offered to evaluate a speaker's position on the topic and verify if the document contains a premise to be classified.

In the following work, firstly, we are going to discuss the works relevant to argument mining for texts about coronavirus. Next, we will use some state-of-the-art algorithms for the RuArg22 dataset. Finally, we implement a condition-based classification pipeline and compare the combination of different models as a part of this pipeline.

2 Literature Review

The covid sentiment analysis has recently become one of the most popular applications of NLP, as many researchers dedicated themselves to the investigation of posts on social media and people's attitudes towards the things brought to our everyday life by the pandemic. Of the most widely discussed phenomena in the recent scientific literature are vaccines. One of such papers is *Insight from NLP Analysis: COVID-19 Vaccines Sentiments on Social Media* by Tao Na et al., 2021, which analyzes posts on Twitter and tries to label the speaker's feelings towards some brand of vaccine to consequently evaluate which categories of people prefer or distaste some vaccine's manufacturer and vaccination as a whole. As the main and only model, the linear discriminant analysis model is applied to the data. We believe that the paper would benefit from the more powerful machine- and deep-learning methods. Besides, the LDA's assumptions such as conditional normal distribution of the predictors or equal covariance matrices for the classes may not be satisfied given the nature of the text features. Despite the fact that work could be significantly improved by other models, we still can incorporate some workflow from the paper and apply similar data pre-processing techniques and text inference. Additionally, the documents in the RuArg22 dataset, that are short and some of which are conversational, are quite close to the messages usually posted on Twitter, so it would be beneficial to this study to address the works analyzing coronavirus on this social media.

Another popular COVID-related NLP application is the detection of fake or untrustworthy news. In such works, we are particularly interested in the "structure-based fake news" detection rather than text-based because it can be applied for sentiment identification as well. We consult with Abdullah Hamid et al. and their *Fake News Detection in Social Media using Graph Neural Networks and NLP Techniques: A COVID-19 Use-case*, 2020. Importantly, the authors apply the most sophisticated NLP techniques for word representation such as BERT embeddings and Bag of Words. The paper describes the neural network architectures that will be applicable for our problem of stance and argument classification.

A similar paper that is of great importance to us is *COVIDLies: Detecting COVID-19 Misinformation on Social Media* by Tamanna Hossain et al., 2020, and considers more than 20 different models while labeling the stance in several thousands of Twitter posts. As in our problem, the researchers attempt to extract a speaker's opinion about some coronavirus topic and observe that BERT embeddings combined with a neural network result in the highest F1-score. Hence, we motivate the choice of one of our models by this article.

3 Data Analysis

The training set for our competition contains 6717 observations with six 4-class target variables corresponding to a speaker's stance and argument for three topics. Interestingly, the messages contain official announcements such as example 1 in table 1, and there are many informal comments-like messages in example 2. Additionally, some of the messages contain inappropriate and offensive lexicon as example 3. As a result, it may lead us to a multi-purpose pre-trained model rather than a context-specific one in order to capture the sentiment in both formal and informal texts.

№	**Original Text**	**Translation**
1	О несоблюдении карантинных мер контактными лицами можно сообщить на на горячую линию…	Non-compliance with quarantine measures by contact persons can be reported to the hotline...
2	[USER], подождите недели две после карантина, не долго осталось!	[USER], wait two weeks after quarantine, not long left!
3	… вот из-за таких идиотов, которые ходят без масок и не сидят на карантине страдают все!	… that's because of such idiots who do not wear masks and do not sit in quarantine, everyone suffers!

Table 1: Examples of texts in the training set.

class		**quarantine stance**	**quarantine argument**	**vaccines stance**	**vaccines argument**	**masks stance**	**masks argument**
-1	**"irrelevant"**	4627	4617	5059	5059	3587	3587
1	**"other"**	1341	1756	866	1238	1832	2451
2	**"for"**	587	217	374	149	704	339
0	**"against"**	172	127	418	271	594	340

Table 2: Value counts of the target columns.

We start with the analysis of the target variable for different topics and observe that the classes are highly unbalanced, with 0 and 2 being the most problematic classes as shown in Figure 1. Unless some measures are taken, all classification models built on such data will be biased towards predicting the majority classes and giving a poorer performance. Therefore, we may need to further consider the methods for target balancing. Another observation is that, for all target columns, class "irrelevant" (i.e. -1) appears approximately as often as the other three classes taken together, meaning that it may be reasonable to consider developing a model that checks for relevance (presence of each topic in the document) first.

4 Data Preparation

As the next step, we need to clean the text starting with a very basic approach. We convert the strings to the lower case, remove special symbols and punctuation. As the organizers of the competition wanted to make the texts impersonalized, they had removed the names or any other references to people's names and replaced it by a certain substring. As a result, the texts contain a special substring "[USER]" that does not entail any special meaning and can thus be removed. The obtained dataset will be further processed and modified by the models or the pipeline.

5 Methodology

We would like to discuss the approaches that were used in the work and compare those methods in terms of prediction performance. We decide to avoid the code from the baseline, implement the methods independently, and not use any additional data from the internet.

Competition Baseline. Based on the *"DeepPavlov/rubert-base-cased-sentence"* embeddings, the same neural network (256-768-4) is fitted with Adam optimizer during 20-epoch to predict two 4-class stance and argument columns simultaneously.
Logistic Regression. The simplest approach in our analysis applies a sigmoid function to the linear transformation of the initial features. Due to its design, the model is good at catching some linear relations when a word affects the target notably but is not advanced enough to capture some hidden patterns.
SVM. Support Vector Machines is a statistical learning method that draws a separating hyperplane in the initial or augmented feature space using a kernel. We choose the sigmoid kernel function as it has proven itself well in many NLP classification works. It is powerful in a binary classification, hence we apply it while labeling a new topic variable as described in the following section.
FastText. FastText is a text classification model developed by the Facebook AI Research lab. Due to the hierarchical structure of the final classifier and decision trees underlying the methods, the computation time is reduced significantly, while the prediction performance is not inferior to other advanced and sophisticated NLP approaches. Importantly, the model contains an internal word embedder allowing us to pass the text without pre-vectorization. The most accurate results were achieved by setting the learning rate to 0.12, number of epochs to 30, the loss function to softmax, and the number of Ngrams to 2.
Neural Networks. One of the most powerful methods in machine learning and NLP, owing to its flexibility and dense layer architecture, is neural networks. We use a 3-layer (768-40-3) fully-connected neural network with ReLu activation function, Adam optimizer, and sparse categorical cross-entropy loss.
Random Oversampling and Undersampling. The oversampling (undersampling) method finds which class has the greatest (smallest) number of observations of the target variable and samples randomly the rows of the dataset to make an equal number of observations for all classes. Due to the fact, the competition dataset is of relatively small size and the minority class for some targets accounts for no more than 3%, the undersampling methods showed poorer performance.
Vectorization Techniques. In this analysis, we consider the following pre-trained word embeddings from the HuggingFace AI repository:

1. *DeepPavlov/distilrubert-tiny-cased-conversational*
2. *cointegrated/rubert-tiny-bilingual-nli*
3. *cointegrated/rut5-small*
4. *cointegrated/rubert-tiny*

Besides, we try count vectorization and tf-idf transformation with logit and SVM.

6 Strategies

We study several strategies and compare the performances using averaged F1 measures for all topics without considering the “irrelevant” class.

FastText with Balancing. Due to the incorporated vectorization technique, it may be reasonable to consider this method as a standalone classifier. We consider each of the six target columns separately and fit the model to label 4 classes.

Multi-Stage Classification with Balancing. We would like to discuss the hierarchical way the models are fitted. For each topic, for example for masks, we fit the first model (M0) to distinguish a new bool variable called *topic* which checks if the stance is negative 1 or not. Thereafter, we fit two other models (M1 and M2) separately on the balanced slice of the dataframe where the topic is not 0 i.e., where the stance is 1, 2, or 3. In such a manner we obtain 9 different models overall. The procedure is shown in Figure 1.

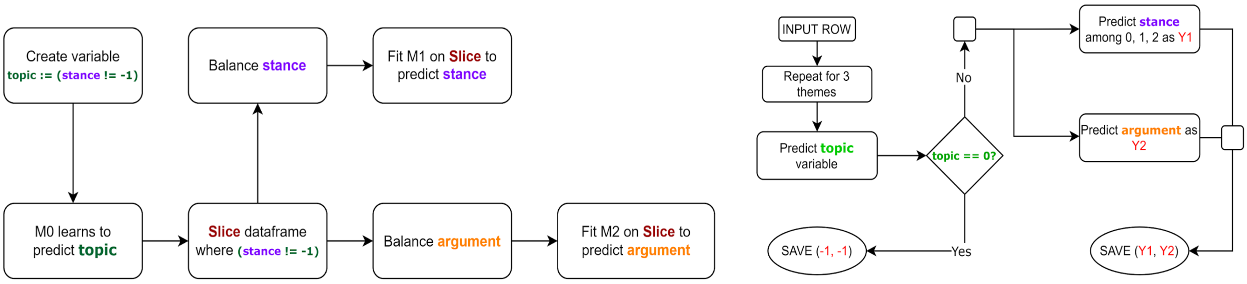

Figure 1: Left: Scheme of the fitting procedure. Right: Scheme of the forecasting procedure for each entry in the test set.

Strategy B can bring us to many variations due to options in the choice of models M0, M1, M2, and the corresponding vectorization techniques. As for the prediction procedure (see Figure 1: Right), we process each entry in the test set using the same approach as for the training data and forecast the value for the topic variable first. If it's 0, we may understand that this topic (like quarantine) was not mentioned in the sentence, so stance and argument are both saved as irrelevant (-1). Otherwise, we apply two separate neural networks to predict balanced stance and argument.

7 Results Discussion

After trying different combinations of models for predicting topic, stance, and argument, we are ready to compare their performances using table 3. We may understand that the result from M0 in the condition-based prediction is the most important as it affects the values of other predictions. Hence, we are mostly concentrating on trying different types of M0 and corresponding vectorization methods.

As we can see from the last column, the labeling premise is the most problematic for all models. While all the approaches successfully pass the stance baseline notably, just a few of them outran the premise baseline significantly. We assume that this is largely due to the fact that the premise was not as dependent on the topic as the stance. Since the powerful FastText model as well as the baseline were outperformed by most pipeline combinations, we consider the idea of extracting the topic variable (consequently reducing the number of classes in the argument variable) quite an efficient approach. We also note that we have attempted each combination without class balancing and observed that the scores drop dramatically by around 0.08-0.10.

The most powerful in terms of predictability appears to be three neural networks with the “*DeepPavlov/distilrubert-tiny-cased-conversational”* word vectorization. Unfortunately, although “cointegrated/rubert-tiny-bilingual-nli” demonstrates one of the highest F1-Premise, none of the “cointegrated/..” embeddings improve the scores.

№	M0	M0-Emb	M1	M1-Emb	M2	M2-Emb	F1 Stance	F1 Premise
0	Baseline						0.392	0.451
1	FastText						0.463	0.462
2	LOGIT	tf-idf	LOGIT	tf-idf	LOGIT	tf-idf	0.430	0.361
3	LOGIT	tf-idf	NN	DeepPavlov	NN	DeepPavlov	0.499	0.525
4	SVM	tf-idf	NN	DeepPavlov	NN	DeepPavlov	0.496	0.494
5	SVM	DeepPavlov	NN	DeepPavlov	NN	DeepPavlov	0.509	0.529
6	**NN**	**DeepPavlov**	**NN**	**DeepPavlov**	**NN**	**DeepPavlov**	**0.530**	**0.559**
7	NN	bilingual-nli	NN	bilingual-nli	NN	bilingual-nli	0.478	0.521
8	NN	rut5-small	NN	rut5-small	NN	rut5-small	0.470	0.484
9	NN	rubert-tiny	NN	rubert-tiny	NN	rubert-tiny	0.495	0.495

Table 3: Performance comparison.

Opportunities for Improvement

We could consider other competitive classification models such as LSTM or gradient boosting. It can be further improved by other COVID-specific tokenizers. The errors and the best classifiers could be analyzed by explanatory approaches such as LIME and SHAP.

References

[1] Kotelnikov, E., Loukachevitch, N., Nikishina, I., & Panchenko, A. (2022). RuArg-2022: Argument Mining Evaluation. In Computational linguistics and intellectual technologies: Papers from the annual conference “dialogue”.

[2] Osipov, G., Panov, A., & Yakovlev, K. (2019). Natural Language Processing with DeepPavlov Library and Additional Semantic Features. Artificial Intelligence. Lecture Notes in Computer Science, vol 11866. Springer, Cham.

[3] Na, T., E., Cheng, W., Li, D., Lu, W., & Li, H. (2021). Insight from NLP Analysis: COVID-19 Vaccines Sentiments on Social Media. Computation and Language; Social and Information Networks.

[4] Hamid, A., Shiekh, N., Said, N., Ahmad, K., Gul, A., & Al-Fuqaha, A. (2020). Fake News Detection in Social Media using Graph Neural Networks and NLP Techniques: A COVID-19 Use-case

[5] Hossain, T., Logan, R., Ugarte, A., Matsubara, Y., Young, S., & Singh, S. (2020). COVIDLies: Detecting COVID-19 Misinformation on Social Media

[6] Joulin, A., Grave, E., Bojanowski, P., Douze, M., Jégou, H., & Mikolov, T. (2017). FastText.zip: Compressing text classification models. 11th International Conference on Information and Communication Systems (ICICS), pp. 243-248

[7] Mohammed, R., Rawashdeh J., & Abdullah, M. (2021). Insight from NLP Analysis: COVID-19 Vaccines Sentiments on Social Media. Computation and Language; Social and Information Networks.