ELMo Word Representations For News Protection

Elizaveta Maslennikova[1]

[1] National Research University Higher School of Economics (HSE) N. Novgorod, Russia
maks_lizok@mail.ru

Abstract. Within framework of the research for this article a new state-of-the-art ELMO model of the words representation to improve the quality of classical models for solving the problem of binary classification in different interpretations is considered. The article contains a description of various methods applied to the processing of source texts for their transformation into the format necessary for many models, a description of their advantages and disadvantages, principles for constructing and operating the context-dependent representation of ELMo with a detailed description of the algorithm for using it within the target model. For a competent assessment of the results obtained, all experiments are carried out using a real dataset including news articles from various sources in China and India. Comparative analysis includes consideration of the results of adding an ELMo model to standard target models of solving a problem in comparison with using Word2Vec. A comparison is also made for different problem statements - the classification of whole texts, individual sentences and the finding of specific passages.

Keywords: ELMo, BiLM, Text Classification, NLP.

1 Introduction

At the present, a person is surrounded by a large number of external sources of information. The newspapers, magazines, radio, TV, even unwittingly heard or seen news can sit in person's head, then reborn into some kind of idea or even a change of own principles. Now, in the sphere of high technologies, the Internet is becoming increasingly popular as a way to communicate, learn, search for information and as a guide to the world of the latest news. At the same time, the world wide web is growing at an incredible speed, the number of sites on various subjects is becoming more and more, news portals fully replace the news program on any channels, and social net-works multiply the number of users every day. But, unfortunately, such a speed of distribution creates great difficulties with the problem of controlling all published content, especially since such a large coverage of the World Wide Web contributes to increased interest from people or their groups who are trying to spread protest ideas, irrelevant or

even prohibited content. Protest news also includes appeals to organize or participate in various events that are in the scope of contentious politics and characterized by riots and social movements, i.e. the “repertoire of contention” (Giugni 1998, Tarrow 1994, Tilly 1984). Therefore, the task of creating a mechanism capable of controlling the published content and identifying protest content, event and information connected with them is a very important and an urgent problem today. Thus, this article is devoted to the study of algorithms for solving the actual problem of binary classification by the presence of some kind of protest ideas in it.

Currently, machine learning algorithms are gaining much popularity in the study of classification problems. At the same time, the most important problem and interesting part of the development is the creation of a model capable of translating a complex human language into a machine-friendly form. language is a very complex structure: letters combined in a different order constitute completely different words, and sometimes the same word has several meanings; words, combining sentences in a different order, can give a completely different emotional coloring, and sometimes a different meaning; etc. At the moment, researchers have proposed a large variety of different models for representing words in an “understandable” form for a computer. But mostly all of them are based on the coding of either the letters that make up the word, or the coding of the words themselves, without taking into account their lexical meaning and the whole context surrounding it. But, unfortunately, in the present conditions, when models for many tasks cannot be trained on a sufficient number of texts, and their writing style contains a very complex structure, these models are not sufficiently accurate and workable. Therefore, in recent years, researchers have been quite actively developing models for the representation of words that combine both syntax, semantics, and lexical meaning of individual tokens. One of such models is ELMo (Embeddings from Language Models) - the representation of words as a vector of features obtained from a neural network pre-trained on a huge text package using LSTM (Long short-term memory) layers. This model was developed and presented by researchers from Washington in 2018, and produced a great resonance, as its effectiveness was proven when applied to some well-known machine learning tasks (Named Entity Recognition, Questions answering on the text, etc.) - the quality of the best models found showed an even higher result when using ELMo as models for the representation of texts for further research than those models that were previously the most effective in their own right. At the same time, it was precisely the task of classifying texts that was not part of their research, although it is expected that the described representation will be able to improve the quality of models for classifying texts, especially in conditions of a small amount of training data set.

In this paper, the possibility of using the ELMo model for the task of recognizing protest ideas in texts is observed. The structure of the creation of the ELMo model, studying of other models of the words representations that were previously used for such tasks; carrying out their comparative analysis and drawing conclusions regarding the possibility of improving the quality of standard algorithms using ELMo is also presented in this work. It is as well necessary to consider the machine learning algorithms

applied earlier to the problems of classification, implement them using the characteristics from ELMo and carry out a comparative analysis with the results obtained without it using on real data sets.

2 Related works

As mentioned earlier, the main problem in building a model for word processing is the choice of a method for converting text into a "understandable" form for a computer. For these purposes people often use various kinds of embedding. Embedding is a process of matching a certain object (text, word, picture, etc.) with a certain vector of numbers. Correspondingly, the source texts, encoded by matching the words to a point in n-dimensional space, take the form of a computer model that can be processed.

One of the most simple and widely used approaches for encoding words with a vector of numbers is the Bag-of-Words method [1], the main idea of which is to form a dictionary of all the words of the source text, organize it, and then convert the texts in the vector of numbers, where the i-th element is equal to the number of occurrences of the i-th word from the dictionary in the given text. This approach gained its fame due to its simplicity and sufficiently large efficiency for processing a small number of texts. With the development of technology and the emergence of the Internet in human life, the number of processed texts and their complexity is growing every day, making models like the BoW inapplicable. Then, in 2013, the Word2Vec model was proposed, which not only is capable of working with a large volume of texts and a huge dictionary, but which only "wins" with the growth of information [2]. This approach is based on the locality hypothesis - words that occur in the text next to identical words will have the close coordinates of words at the output, i.e. it is assumed that only words that are combined with each other can stand next to them. However, these approaches allow only one context-independent representation for each source word. A little later, several more models were proposed that try to circumvent this drawback by examining individual vectors for each word value [3] or by enriching the original word with information on its subwords (using a letter-by-word representation) [4].

Another quite popular word representation model is Context2Vec [5], which uses a bidirectional network of long short-term memory [6] to encode a context around a word. Another approach to the study of contextual embedding includes the keyword itself in the presentation and is calculated, for example, using controlled neural machine translation [7]. Like the previous approaches, these models only "win" from a large amount of input data.

Previous studies on the topic of machine learning words processing also showed that different layers of deep bidirectional recurrent neural networks are capable of encoding various types of information. For example, the introduction of multitasking syntactic control (using the tags of parts of speech) at lower levels of LSTM can improve the overall performance of higher-level tasks [8]. Long Short-Term Memory – is the kind of recurrent networks which is capable of learning long-term dependencies, while mem-

orizing and transmitting some information for long periods of time is their usual property, and not something that they hardly try to learn as with the simple architecture of recurrent neural network.

In the machine translation system, also based on recurrent networks, it was shown that the representations obtained at the first level in the two-layer LSTM predict tags of parts of speech better than at the second level [9], and, finally, the upper level of LSTM for encoding is studying the meaning of the word [5].

The described below ELMo model of the word's representation have all the advantages of considered models and in which their shortcomings will be also taken into account [10].

3 Embeddings from Language Models

The model of the word's representation studied in this article differs from the traditional ones in that each token is assigned a representation, which is a function depending on the entire input sentence. In this case, I use vectors derived from a bi-directional network of long short-term memory, which is taught in advance on a large text package as a model representing the entire language used. Therefore, this approach received such a name: Embeddings from Language Models.

Let we have a sequence of N tokens $(t_1, t_2, \dots, t_N)$. The forward language model (LM) calculates the probability of a given sequence, simulating the probability of the appearance of the token t_k taking into account the history $(t_1, t_2, \dots, t_{k-1})$:

$$\boldsymbol{p}(t_1, t_2, \dots, t_N) = \prod_{\boldsymbol{k=1}}^{\boldsymbol{N}} \boldsymbol{p}(t_k | t_1, t_2, \dots, t_{k-1}) \quad (1)$$

Recent most popular language models compute a context-independent representation of tokens x_k^{LM} (using, for example, convolutional neural networks above the letters that make up the original text), and then pass them through L layers of the forward-directed LSTM. It turns out that for each position k, each layer of the LSTM network outputs context-dependent representation $\underset{h}{\rightarrow}_{k,j}^{LM}$, where j = 1, ..., L and the output of the upper layer of this network $\underset{h}{\rightarrow}_{k,L}^{LM}$ is used to predict the next token t_{k+1} (calculating the Softmax activation function).

The backward language model is similar to the forward language model (1), with the only exception that the passage through the sequence is carried out in the reverse order, predicting the previous token, taking into account the subsequent context:

$$\boldsymbol{p}(t_1, t_2, \dots, t_N) = \prod_{\boldsymbol{k=1}}^{\boldsymbol{N}} \boldsymbol{p}(t_k | t_{k+1}, t_{k+2}, \dots, t_N) \quad (2)$$

The implementation occurs by analogy with the forward language model, where on each reverse LSTM layer j= 1, ..., L give a context-sensitive representation $\overleftarrow{h}_{k,j}^{LM}$ for token t_k, taking into account the following sequences $(t_{k+1}, t_{k+2}, \dots, t_N)$.

Accordingly, biLSTM combines these approaches for both forward (1) and backward (2) language models. In this presentation, the logarithmic probability will be jointly maximized taking into account both directions:

$$\sum_{k=1}^{N}\left(\log p\left(t_k \middle| t_1, \dots, t_{k-1}; \Theta_x, \overrightarrow{\Theta}_{LSTM}, \Theta_s\right) + \log p\left(t_k \middle| t_{k+1}, \dots, t_N; \Theta_x, \overleftarrow{\Theta}_{LSTM}, \Theta_s\right)\right) \quad (3)$$

where Θ_x - the representation of the token, Θ_s - the result of applying the SoftMax layer, $\overrightarrow{\Theta}_{LSTM}$ and $\overleftarrow{\Theta}_{LSTM}$ are the outputs after the LSTM layer, taking into account the previous and subsequent context respectively. The schematic representation of the architecture of the bi-directional language model using in ELMo is presented in Fig.1. Embeddings from Language Models is a context-dependent specific to the task model of representing words, which is a combination of representations from the intermediate layers of the biLSTM network. For each token t_k a network of depth L gives a total of 2L + 1 different representations (outputs from all layers):

$$R_k = \left\{x_k^{LM}, \overrightarrow{h}_{k,j}^{LM}, \overleftarrow{h}_{k,j}^{LM} \middle| j = 1, \dots, L\right\} = \left\{h_{k,j}^{LM} \middle| j = 0, \dots, L\right\} \quad (4)$$

where $h_{k,0}^{LM}$ – input layer, $h_{k,j}^{LM} = \left[\overrightarrow{h}_{k,j}^{LM}, \overleftarrow{h}_{k,j}^{LM}\right]$ - representation from the j-th layer of biLSTM taking into account both directions, j = 1, ..., L. The overall final ELMo representation, which is "embedded" in the main model to solve the NLP problem, is a convolution of outputs from all layers or all vectors from R into one vector: $ELMo_k = E(R_k, \Theta_e)$. In the simplest case, you can use the output from the entire network (the presentation from the last layer) $E(R_k) = h_{k,L}^{LM}$, but then the representation of the words will be based only on the language model that we assume not quite accurate, and will not depend on the current problem being solved and it's training dataset. If we approach the problem more globally, then we can calculate the final presentation as some combination of outputs from each layer with the corresponding weights, which just will be selected in the process of learning the final model:

$$ELMO_k^{task} = E(R_k, \Theta^{task}) = \gamma^{task} * \sum_{j=0}^{L} s_j^{task} h_{k,j}^{LM} \quad (5)$$

where the parameters s^{task} (normalized weights vector) and γ^{task} (scalar parameter which is necessary to assist the optimization process) allow the entire ELMo vector to be scaled within a specific task (see Fig.2).

Having a pre-trained biLM (Bidirectional Language Model) with the architecture described above and a certain algorithm for solving the target problem of NLP it is sufficiently easy to integrate the ELMo model into the existing final solution to improve it. For this, it is enough to run the biLSTM network on the available source data and save all views from each layer of this network, and "train", or rather, select the necessary weights for these models while training process using the algorithm described below.

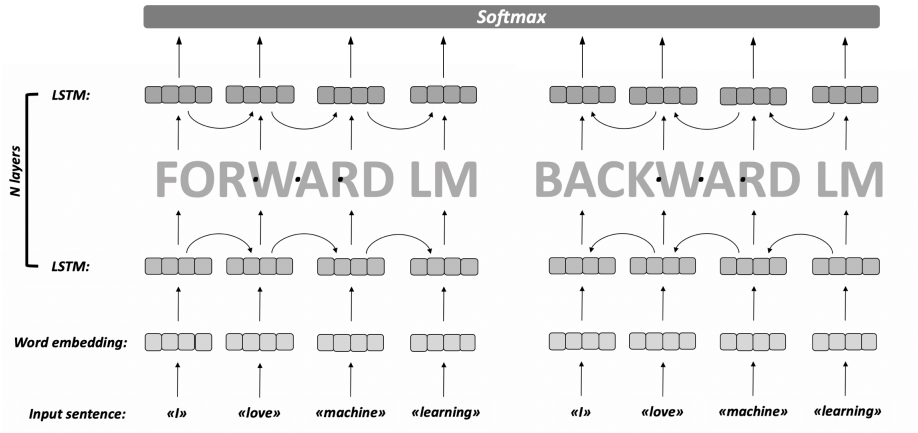

Fig. 1. Architecture of the bi-directional language model using in the ELMo.

First of all, you need to consider the lowest layer of the original model. For most tasks in NLP, the source data has a similar structure, and, accordingly, similar data processing and architecture on the very first layer of the model, which makes the algorithm for adding the ELMo model quite universal. The standard way of processing the original sequence of tokens $t_1, \ldots, t_N$ is applying to it some algorithm for generating a context-independent representation of the word x_k (possibly using a previously trained network or based on a symbolic representation). Then the data is transferred further.

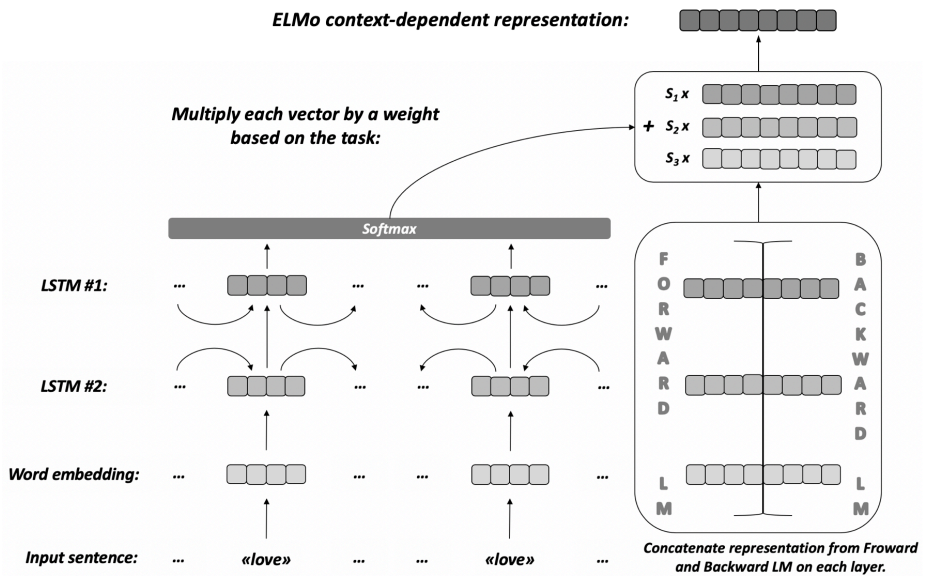

Fig. 2. Example of the formation of the final presentation for the word in two-layer ELMo.

For adding ELMo to the final model, it is necessary to pass the initial representation of the token x_k through the pre-trained biLM, and then send the received ELMo vector for training to the next layers of the original model. For some tasks using the $[x_k, ELMo_k^{task}]$ as the final representation will be more efficient, but for the problem studied in this article, with a very small initial training dataset, this representation of words greatly complicates the model, but not gives a higher result because the data is not enough for full-fledged training.

As previously described, I use a pre-trained on a huge corpus of text data (1 Billion Word Language Model Benchmark) network as a biLM in this work.

The final model for pre-trained biLM is a recurrent neural network with two biLSTM layers with 4096 and 512 dimensions respectively, and a residual connection between the first and second layers (i.e., after two layers, a representation is made up as the sum of from the 1st and from the 2nd layer). At the same time, the model is built in such a way as you can use not only the final ELMo output, but also get separately outputs from each layer of the network or immediately get a general representation for some set of tokens (using the convolutional mean-pooling layer). After this training and such setup, we have finished language model, which can be incorporated into the final model for most tasks from NLP to improving the most effective (State-of-the-art) methods. It should be noticed, that for some tasks, the ELMo model, even without adjusting the weights for the outputs from each layer, provides an increase in quality, especially if it is close to a set of training data for the final task for teaching a language model. In the scope of this article, I explore the application and effectiveness of the ELMo model with the SVM classifier, fully connected, convolutional and recurrent neural networks.

4 Experiments

Experimental dataset is a set of texts taken from various English-language news sources from India and China. For different levels of complexity, the data is presented in a different format:

• Task 1 - the classical formulation of the original problem, in which, according to a set of news articles, it is necessary to determine protest event related news articles as a whole or any other news article;

• Task 2 is a more difficult task of binary classification, where a whole set of texts is also presented, but each of their sentences should being considered separately in the context of having a protest event trigger or a mention of it;

• Task 3 - this problem formulation is similar to the task of Named Entity Recognition. The main goal is to extract the event information that targets protest event.

Data package for all tasks is represented a set of text objects with an assigned class for each. The difference lies only in the object itself - this is either the whole text, its separate sentences or separate words. It should also be noted that only data from Indian resources are submitted for training, and data from both countries are presented for testing. This is necessary for a higher test of the possibility of generalizing the models, since it is assumed that the test data from India has the same distribution as the training data, since the distribution of the data from China should slightly differ [11].

As it was noted earlier, for correctly testing of ELMo model performance, it is necessary to compare the result of the work of the target model for solving this problem using the ELMo word representation and without it (using, for instance, the Word2Vec model). At the same time, it is necessary to choose the final model correctly, which would be suitable within the context of the task and would give good results even without using ELMo. Therefore, in the framework of the experiments, several classifiers were used to solve this problem with the selection of all the necessary parameters. The final used architecture of fully connected network contains 4 Dense layers for Task 1 and 6 these layers for Task 2 and 3, the best architecture of convolution network in the scope of these task is a combination of Convolution and MaxPooling layers repeated 3 times, GlobalAveragePooling, Dropout and Dense layers and the final used recurrent network consists of 3 biLSTM layers. These architectures were chosen as the best of all reviewed after a series of experiments. All necessary parameters as the number of neurons on hidden layers, units in LSTM were selected using the grid search method for each task individually.

Table 1. The results of applying different classifiers with ELMo for described tasks.

	Task 1		**Task 2**		**Task 3**	
Classifiers	***India***	***China***	***India***	***China***	***India***	***China***
SVM + ELMO	79.13	57.30	61.17	55.42	-	-
FullyCon.NN + ELMo	**79.37**	**60.00**	64.93	59.11	33.46	21.72
Convol.NN + ELMo	79.07	59.84	65.39	**64.86**	35.78	25.16
Recur.NN + ELMo	73.12	54.67	**65.54**	63.92	**52.40**	**42.50**

As can be seen from obtained results (see Table 1), SVM is a fairly good model for binary classifying whole texts, applying it to test data from the same distribution. But with respect to data from the second test set, SVM gives worse results than neural networks, which shows a poor generalizing ability of this model. Therefore, the most successful model for solving this problem in a general sense is the fully connected multilayer fully connected, which shows the best result for the Chinese data set and high result for the Indian. It should be noted right away that the SVM classifier is not suitable for other tasks, since it does not take into account the previous context in any way but treats each object as an independent unit. This is also confirmed by the rather low result of its application to Task 2. Moreover, for example, the recurrent network is not effective for Task 1, since, on the contrary, it tries to use the previous context, while each object is a separate unit under consideration in this task. Convolutional and recurrent neural networks showed good results, while the generalizing ability is better for the first one for task 2. As for Task 3, there, as it was expected, the recurrent network gives the best quality, since individual words do not represent the most meaningless context, especially if parts of information of protest news can include several words in a row.

As it was already mentioned, the data presented in the Table 1 represent the result of applying various target classifiers using ELMo model. At the same time, this model is

built into neural networks with the ability to configure weights for output from its different layers, but SVM should get representation obtained after passing through ELMo with fixed weights installed during model formation.

To prove the effectiveness of the using ELMo model the Fig. 3 provide a comparative analysis of the use of various application models with the Word2Vec and ELMo presentation. It is clearly seen that the ELMo model gives an increase in the quality of recognition for all models (from 1% to 10% of F1 score), this is especially clearly seen in Tasks 2 and Task 3. From the obtained result, we can say with confidence that the ELMo model is really able to improve the results of the classical models for solving the problem of binary classification. That is why, after its invention, this model received the status of the-state-of-the-art very quickly. Its effectiveness is also confirmed by the fact that the solutions presented in the framework of this article at the CLEF Protest-News 2019 competition took high prizes [12].

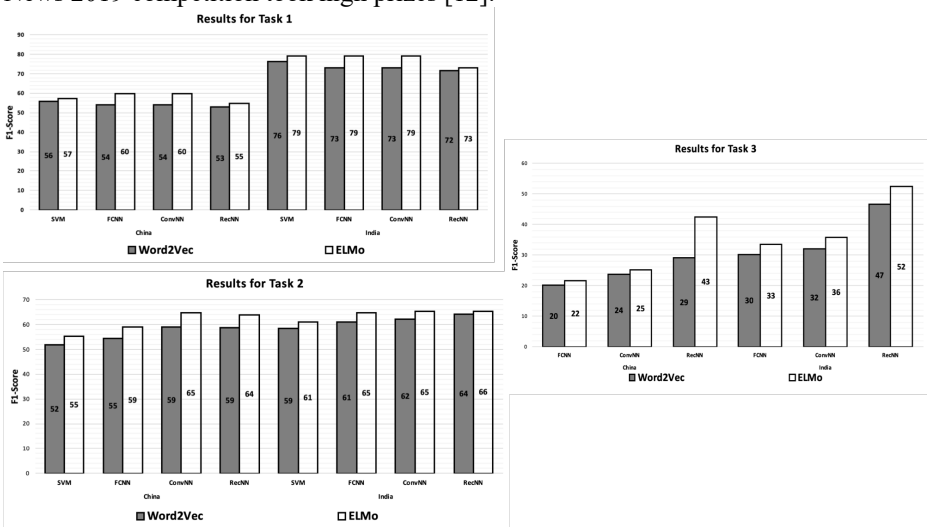

Fig. 3. Comparative analysis of using SVM, Fully Connected, Convolution and Recurrent Neural Networks for binary classification with ELMo and Word2Vec words representations.

5 Conclusion

This article examined the possibility of using the ELMo word representation model to improve the quality of prediction of classical models for the problems of binary classifying according to the presence of protest ideas or information about it in them. Within the framework of the research, the predecessors of ELMo, the principles of its construction and operation, as well as the possibility of its introduction into classical models of problem solving in NLP were considered. After the experiments and analysis of obtained results, it is safe to say that ELMo really improves the quality of many models for solving text analysis problems in comparison with the use of other word representation algorithms in a model-friendly way. The ELMo method really deserves the title of state of the art at present. installed during model formation.

As a future work I plan to explore effectiveness of ELMo model with different methods of its training previously, using different datasets. Moreover, the study of the possibility of combining different pre-trained language models is also included in the scope of my further research.

References

1. Harris, Z.: Distributional Structure. Word 10(2-3), 146-162 (1954).
2. Mikolov, T., Chen, K., Greg, C., Dean, J.: Efficient Estimation of Word Representations in Vector Space. In: Proceedings of the International Conference on Learning Representations (2013).
3. Neelakantan, A., Shankar, J., Passos, A., McCallum, A.: Efficient non-parametric estimation of multiple embeddings per word in vector space. In: EMNLP (2014).
4. Wieting, J., Bansal, M., Gimpel, K., Livescu, K.: Charagram: Embedding words and sentences via character n-grams. In: EMNLP (2016).
5. O. Melamud, J. Goldberger and I. Dagan, "CoNLL," in context2vec: Learning generic context embedding with bidirectional lstm, 2016.
6. Hochreiter, S., Schmidhuber, J.: Long Short-term Memory. Neural Computation 9(8), 35-80 (1997).
7. McCann, B., Bradbury, J., Xiong, C., Socher, R.: Learned in translation: Con- textualized word vectors. In: NIPS (2017).
8. Hashimoto, K., Xiong, C., Tsuruoka, Y., Socher, R. : A joint many-task model: Growing a neural network for multiple nlp tasks. In: EMNLP (2017).
9. Belinkov, Y., Durrani, N., Dalvi, F., Sajjad, H., Glass, R.: What do neural machine translation models learn about morphology? In: ACL (2017).
10. Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L.: Deep contextualized word representations. In: NAACL (2018).
11. CLEF PROTESTNEWS 2019, https://emw.ku.edu.tr/clef-protestnews-2019/, last accessed 2019/06/01
12. CLEF 2019 Lab ProtestNews, https://competitions.codalab.org/competitions/22349#results, last accessed 2019/06/01