Linguistic parameters and word embeddings for protest news detection in text.

Chedi Bechikh Ali[1]

LISI laboratory, Université de Carthage, Tunisie
chedi.bechikh@gmail.com

Abstract. We present in this paper our participation in ProtestNews lab at CLEF 2019 in task 1 and task 2. In task 1, the objective is to predict if an article contains protest news or not. In task 2, we must decide if a sentence contains a protest event or not. For these two tasks, we used a supervised machine learning approach based on the logistic regression model. We combine the supervised learning algorithm with two different natural language techniques. The first relies on text processing with linguistic properties. The second is based on the expansion of the text with related term using word embedding similarity.

Keywords: Linguistic parameters · compound nouns · word embeddings · supervised learning.

1 Introduction

This paper describes the participation of LISI laboratory at the Conference and Labs of the Evaluation Forum (CLEF) 2019 ProtestNews for the detection of a protest event in news articles. We submitted results obtained from different approaches. In this paper, we describe the different proposed approaches as well as the findings concerning the results. The ProtestNews lab proposed three tasks:

- Task 1 is a classification task, it consists to identify which text contain protest news.
- Task 2 objective is to classify if a sentence contains and event-trigger of protest or not.
- Task 3 is an information extraction task, the objective is to extract locations, participants and time about protest event.

The reminder of this paper is organized as follow, in section 2 we describe our methodolgy based on supervised classification algorithm. In section 3 we describe the linguistics characteristics that we use to process the documents. Then, in section 4 we present our document expansion approach. In section 5, the experimental results are presented. Then we conclude and present future works.

Copyright © 2019 for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0). CLEF 2019, 9-12 September 2019, Lugano, Switzerland.

CEUR-WS.org/Vol-2380/paper_115.pdf

2 Methodology

This section describes the model that have been used to classify the test data of both task 1 and task 2. The model is applied for both tasks, and use the same set of linguistic settings.

The overall architecture of our proposed framework consists of two main phases: training and testing. In the training phase, the classifier learns from a set of labeled text. Subsequently, the classifier is capable of classifying new unlabeled documents in the testing phase.

Each phase consists of the following steps: documents preparation, features extraction/selection, and classification. We opt for a supervised approach based on a classification algorithm [5]. The classification algorithm was implemented using the scikit learn[1] library which is a machine learning library for the Python programming language.

We notice that ProtestNews documents present different characteristics for each task: documents in task 1 are composed of long sentences, this may cause drift in the classification process. In the other side, documents for task 2 are short, so they don't contain enough context for the training step. This can lead to different problems: word ambiguity and word mismatch between training data and test data.

To deal with these two tasks, we compared different classification algorithms, among them logistic regression algorithm, random forest, and naive Bayes algorithm. Preliminary experiments were carried on the development data after a training step. Based on these finding, we decided to use the logistic regression algorithm for the rest of the experiments because it gives the best results.

3 Linguistic preprocessing

Before extracting the feature vectors it is required to pre-process the data with stop words removal and text lemmatization. We rely on linguistic processings since they lead to good results in previous work for sentiment analyis task [4].

- Stop word removal: We used an English stop words list provided by the Terrier information retrieval team of the School of Computing Science of Glasgow University. The list contains 733 stop words.
- Lemmatization: We have chosen to lemmatize document words to treat the morphological variations and thus to increase the recall. Lemmatization allows transforming words into a reduced form that is the lemma, which leads to ignoring variations in number and gender. We rely on the part-of-speech tagger Treetagger[2] to lemmatize the text.
- Eliminating named entities (person, place, organization) form text content, because they can't represent protest news. Named entity lead to a drift in the classification process because they can present in both protest news and

[1] https://scikit-learn.org
[2] https://www.cis.uni-muenchen.de/ schmid/tools/TreeTagger/

regular news. In this study, we define named entities as the words that were annotated with the tag "NP" by Treetagger.
- Compound noun (CN) annotation: CN can capture important concepts in the content of the document such as event or concepts related to protest event (e.g.: district violence, armys action, tense situation, etc) [1]. To ensure that we extract CN related to protest events, we first extract CN from documents that are classified as protest news in the training set. Then, we annotate each corresponding CN in the test data. In this work, we only use CN composed of tow words, because in preliminary experiment yield better results than longer CN.

After text processing, we proceed to feature extraction, The aims of the feature selection technique are to find the most relevant features for the classification task. We used unigram because it gives the best results. We didn't rely on any weighting scheme, because using the tf.idf scheme degraded the classification performances in preliminaries experiments with the development data.

4 Document expansion with word embeddings

To deal with the term mismatch problem, we decided to expand documents with the most similar word for each token. Since in previous work using pretrained word embeddings has proven to have a postif impact on different natural language processing tasks: Word Sense Disambiguation, Relational Similarity, Semantic Relatedness [2]. We pose the hypothesis that adding similar or related terms can help to enhance the recall and so the overall performance of the classification process.

For this approach we trained two word embeddings [3]:
- wiki-emb: a word embeddings trained on text8[3] dataset which is a sample of a Wikipedia dump[4] .
- protest-emb: a word embeddings trained on the India training dataset given for task 1.

We choose to train two word embeddings with different data sets, to see if a specialized dataset have an impact on the classification performance, in comparison with text8 dataset which is a 100 megabytes cleaned dataset. For the word embeddings training, we rely on the Gensim python library.

5 Experiments and results

We studied the performances of the proposed approach and we performed different experiments using different setting and processing:

- Run1: consist of applying lemmatization and stop words removal on the training and test set.

[3] http://mattmahoney.net/dc/textdata.html
[4] English Wikipedia dump on Mar 3, 2006

– Run2: consist of combining lemmatization, stop words and named entities removal on both training and test data sets.
– Run3: expanding every word in the sentences with the most similar word from the protest-emb word embeddings.
– Run4: expanding every word in the sentences with the most similar word from wiki-emb word embeddings.
– Run5: combining run 2 settings with the annotation of all CN.

Table 1 present the official submitted runs, there are some runs where there are results only for task 2. The analysis of the results shows that the first run (lemmatizing and eliminating stop word) allows to obtain 0.7612 for task 1 and it corresponds to our second best run. The best result for task 1 was achieved when we expanded the content of the text with bigram extracted from the same span of text. This run allows to obtain the best overall results, but the best result for task 1 (China and India) and the best result for task 2 (India). We note the degradation of the results for task 2 (China).

The best overall result for task 2 is obtained by a simple approach that consists of lemmatizing the text and eliminating stop words.

In a preliminary study phase with development data, we found that expanding text with the most similar word is only beneficial for task 2 and it degrades results for task1. We decided to study the impact of this approach only for task 2. We notice that the best overall results are obtained when training the word embedding on the training set. The best result for task 2 is obtained with word embedding trained on the same data, but the best result for China data is obtained when we used word embedding from another general corpus. This can be explained by the fact that the text8 dataset is bigger dataset and contain more tokens than task 1 training dataset.

Table 1. Classification results for task 1 and task 2 based on F1 measure

model	task1_test	china_task1	avg_task1	task2_test	china_task2	avg_task 2	avg_task
Run1	0.7612	0.3846	0.5729	0.5657	**0.4788**	**0.5223**	0.**5476**
Run2	0.7612	0.4418	0.6015	0.4727	0.3960	0.4343	0.5179
Run3	-	-	-	0.5692	0.4615	0.5150	-
Run4	-	-	-	0.5748	0.4143	0.4945	-
Run 5	**0.7676**	**0.5032**	**0.6354**	**0.5877**	0.3086	0.44819	0.5418

Our official final run was the Run6 and it was ranked sixth among 12 teams. We can note that with this run we achieved the best our results in task 1 and task 2 with India data. Degradation of the performance has been noticed for task 2 with China data, this can be explained by terms mismatch between CN in the training set with india data and the CN with the China data because CN in India data represents other concepts than those extracted in China dataset.

6 Conclusion

This paper describes our participation in the ProtestNews detection lab at CLEF 2019. The aim of this work is to make a decision if a text contains protest news or not. The objective is to develop text classification tools. For this purpose, we used a classifier based on the Logistic regression algorithm. As the first step, we processed the linguistic data processing as a first step. Then, we use word embeddings to expand text with the most similar word. Also, we proposed to add CN extracted from the same text. This work is still in progress and needs more investigations. For future work, we plan to use deep neural network since it achieved good results for other NLP tasks.

References

1. Bechikh-Ali, C., Haddad, H., Slimani, Y.: Empirical evaluation of compounds indexing for turkish texts. Computer Speech & Language **56**, 95–106 (2019)
2. Li, J., Jurafsky, D.: Do multi-sense embeddings improve natural language understanding? In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015. pp. 1722–1732 (2015)
3. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems: 27th Annual Conference on Neural Information Processing Systems. Lake Tahoe, Nevada, United States. December 5-8, 2013. pp. 3111–3119 (2013)
4. Mulki, H., Ali, C.B., Haddad, H., Babaoglu, I.: Tw-star at semeval-2018 task 1: Preprocessing impact on multi-label emotion classification. In: Proceedings of The 12th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT, New Orleans, Louisiana, June 5-6, 2018. pp. 167–171 (2018)
5. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up? sentiment classification using machine learning techniques. In: Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing, EMNLP 2002, Philadelphia, PA, USA, July 6-7, 2002 (2002)