Classification and Event Identification Using Word Embedding

Anaïs Ollagnier[1][0000−0002−4349−5678] and Hywel Williams[1][0000−0002−5927−3367]

Computer Science, University of Exeter, Exeter EX4 4QE, UK
{a.ollagnier,h.t.p.williams}@exeter.ac.uk

Abstract. This paper presents our contribution to the CLEF 2019 Protest-News Track, which aims to classify and identify protest events in English-language news from India and China. We used traditional classification models, namely, support vector machines and XGBoost classifiers, combined with various word embedding approaches. Multiple models were tested for experimental purposes, in addition to the two models evaluated within the official campaign. Results show promising performance, especially in terms of precision on both document and sentence classification tasks.

Keywords: Data mining · Classification · Event detection · Word embedding.

1 Introduction

The CLEF ProtestNews Track was introduced in 2019 aiming to evaluate methods for event classification and detection from news articles across multiple countries. This track has two main goals: firstly, development of generalisable methods which can be applied to heterogeneous news article data; and secondly, to support surveys conducted in other scientific fields such as social and political studies by providing data on political conflict events (e.g. protests, riots). This track includes three tasks: news article classification, event sentence detection and event extraction. Our contribution is focused on the first two tasks. The news article classification task consists of identifying news articles associated with political conflicts through a binary classification scheme (”protest” vs. ”non-protest”). The event sentence detection task focuses on identifying and labeling sentences that refer to protest events (e.g. riots, social events).

Both of the tasks attempted here relate to text classification and sentence classification. Recent work in natural language processing (NLP) and text mining shows many applications that leverage text classification at different levels of scope. At the document level, many classification techniques have been proposed and have achieved good results in the literature [4]. Logistic regression (LR) and support vector machines (SVMs) are two of the most-used techniques [2]. Recently ”deep learning” models based on neural networks have become increasingly popular [3]. At the sentence level, classification must operate on texts that are much shorter than most documents ($\leq$ 160 words), which reduces performance of traditional text classification algorithms. Main limitations concern the feature sparsity of short text which reduces the accuracy of traditional algorithms, such as the similarity algorithm based on word frequency and co-occurred words [1]. To tackle problems arising from short texts, various methods have been proposed to improve their capacity of semantic expression [5]. More recently, NLP has drawn attention in this context through the use of language models learned by word embeddings, especially in models based on neural networks [6,7].

At both document and sentence levels, effective feature extraction is important to help the accuracy and robustness of classification models. Inspired by recent work in efficient word representation learning [9,8] and considering the topical scope of the proposed tasks CLEF ProtestNews Track 2019, here we propose two models that were submitted for official evaluation and also several other models that were developed for experimental purposes. All of them are based on word vector learning combined with linear classifiers. The main aim of these approaches is to find the most efficient feature extraction and classification method which can be applied at different levels of scope.

The rest of this paper is organized as follows. Section 2 presents an overview of the analytical framework. In Section 3, we describe a set of 9 different models using different kinds of word embedding and classifier, with and without dimension reduction. Then, we present results from experimental testing of these models (Section 4) before giving results for the two models submitted to the official CLEF ProtestNews track evaluation (Section 5) .

2 Overview of the proposed framework

In this section, we present the proposed framework consisting of three parts: data processing, word vector learning and text classification.

2.1 Data processing

For each task respectively documents and sentences were converted to lowercase, all URLs and stop-words were removed. After the tokenization process, all tokens based only on non-alphanumeric characters and all short tokens (with < 3 characters) were also deleted. Then, we perform a morphological analysis of all tokens in order to identify lemmas, replacing each token by its lemma.

2.2 Word vector models

In word vector representations, each word is represented by a vector which is concatenated or averaged with other word vectors in a context to form a resulting vector which is used to predict other words in the context [10]. These vectors allow capture of hidden information about a language, like word analogies or semantic associations. In the literature, word vector representations have demonstrated efficiency in boosting accuracy of classification models. However, inconsistent performances are observed in some application contexts [11]. In this paper, we explore three popular embedding models, namely, Word2Vec, GloVe and FastText. Below, we introduce briefly their principle.

- **Word2Vec** [14] is a group of related models based on two-layer neural networks that are trained to reconstruct linguistic contexts of words. Two model architectures can be used: continuous bag-of-words (CBOW) or continuous skip-gram (SG). In CBOW architecture, the model predicts the current word from a window of surrounding context words. As in other bag-of-words approaches, the order of context words does not influence prediction. In the continuous SG architecture, the model uses the current word to define the surrounding window of context words. The SG architecture weights nearby context words more heavily than more distant context words.
- **GloVe** [13] (global vectors for word representation) allows the user to obtain word vector representations by mapping words into a meaningful space where the distance between words is related to semantic similarity. Training is performed on aggregated global word-word co-occurrence statistics from a corpus, and the resulting representations showcase interesting linear substructures of the word vector space.
- **FastText** [12] is based on the SG model, where each word is represented as a bag of character n-grams. A vector representation is associated to each character n-gram; words being represented as the sum of these representations.

Pre-trained word embeddings on large training sets are publicly available, such as those produced for word2vec [14], GloVe [13] or Wiki word vectors for FastText[1].

2.3 Linear classifiers

Despite the popularity of models based on neural networks, linear classifiers stand as strong baselines for text classification problems. Furthermore the state-of-art about these models has proved their suitability and their robustness when they are combined with right features [15]. In addition, neural network models tend in practice to increase computational cost. Following empirical studies conducted on the training set provided for each CLEF ProtestNews task, SVM and XGBoost provided best performances in term of accuracy and log loss scores.

[1] `https://fasttext.cc/docs/en/pretrained-vectors.html` Date of access: 16th May 2019.

3 Proposed models

Nine models were explored for the news article classification and event sentence detection tasks. These models were selected from various combinations within the framework presented above. The best models were chosen according to their global performance in terms of precision, recall and F1-score obtained on the training sets provided for each task. Parameter tuning was performed using GridSearchCV[2] in order to select parameter values that maximize the accuracy of each model. Top parameters are presented in tables following the description of each model below. The model architectures described below were used in similar ways for the two tasks (except for the sum_ner model, see below).

- The **xgboost_fast** model uses word vector representations created by FastText. Vectors were built from the training set provided for each task. Then the XGBoost classifier was used to identify the class of each input.

Data	fraction of columns	Gamma	tree max. depth	min. sum of weights	alpha	fraction of observations
Document	0.75	0.4	5	6	0.005	0.8
Sentence	0.75	0.4	6	6	0.001	0.85

- The **xgboost_fast_wiki** model uses the same architecture as the xgboost_fast model except for word vector learning, which is performed through the use of pre-trained word embeddings. The pretrained model[3] is composed of 1 million word vectors trained on Wikipedia 2017, UMBC webbase corpus and statmt.org news dataset. These vectors in dimension 300 were obtained using the skip-gram model described in [12] with default parameters.

Data	fraction of columns	Gamma	tree max. depth	min. sum of weights	alpha	fraction of observations
Document	0.85	0.1	6	10	0	0.8
Sentence	0.8	0.3	5	12	0.05	0.75

- The **svm_fast** model uses word vectors built from FastText. Vectors were designed from the training sets provided. Then SVM classifiers were used to identify the class of each input.

Data	C	Gamma	Kernel
Document	10	1	rbf
Sentence	0.001	0.001	linear

- The **svm_fast_wiki** model uses classifiers based on SVM. Word vector representations are built from the same pre-trained model that was used for the xgboost_fast_wiki model.

[2] https://scikit-learn.org/stable/modules/grid_search.html Date of access: 16th May 2019.

[3] https://dl.fbaipublicfiles.com/fasttext/vectors-english/wiki-news-300d-1M.vec.zip Date of access: 16th May 2019.

Data	C	Gamma	Kernel
Document	1	1	rbf
Sentence	1	1	rbf

- The **xgboost_glove** model uses a pre-trained word vector embedding as initialization for the representation of words. This embedding named GloVe[4] is composed of 300-dimensional vectors trained over a larger vocabulary of web data (840B words). Then XGBoost classifiers were used to identify the class of each input.

Data	fraction of columns	Gamma	tree max. depth	min. sum of weights	alpha	fraction of observations
Document	0.8	0.0	6	8	0.05	0.75
Sentence	0.75	0.3	6	6	0.05	0.8

- The **xgboost_w2v** model is designed from a word vector representations performed by Word2vec. Vectors were built from the training set provided for each task. Then XGBoost classifiers were used to classify inputs.

Data	fraction of columns	Gamma	tree max. depth	min. sum of weights	alpha	fraction of observations
Document	0.8	0.0	6	12	0.001	0.8
Sentence	0.85	0.3	6	12	0.001	0.85

- The **sum_ner** model uses slightly different text processing according to the application context.
  - For Task 1, the text was trimmed to capture sentences that are most representative of the source document. In this way, we aimed to gain topical clarity and reduce the vocabulary space. Similar to a text summarization process, each sentence was scored as the sum of the weighted frequencies of its words within the whole document. The highest-scoring sentences were then chosen to give a concise representation of the document. The best performances were observed by keeping the first 4 sentences with the highest scores.
  - For both Task 1 (using sentences derived from document level as above) and Task 2 (which begins at sentence level), we then apply text normalization using a named entity recognition tool[5]. Only entities referring to a person, a location or an organisation are identified. Each entity localized is replaced by the name of its class. The aim with this process is to provide harmonized vector patterns which can be beneficial in word representation processes.
  - For both tasks, the final step is to perform classification using the XGBoost technique.

[4] `http://nlp.stanford.edu/data/glove.840B.300d.zip` Date of access: 17th June 2019.

[5] `https://nlp.stanford.edu/software/CRF-NER.shtml` Date of access: 17th June 2019.

Data	fraction of columns	Gamma	tree max. depth	min. sum of weights	alpha	fraction of observations
Document	0.85	0.1	4	12	0.01	0.85
Sentence	0.85	0.4	6	8	0.001	0.8

- The **xgboost_fast_SVD** and **xgboost_fast_wiki_SVD** models were created from the xgboost_fast and xgboost_fast_wiki models above by the addition of dimension reduction alongside feature extraction. Dimension reduction was applied using the Singular Value Decomposition (SVD) method, a commonly applied technique, in order to reduce noise and increase model stability. Briefly, SVD is a matrix decomposition method for reducing a matrix to its constituent parts, to make certain subsequent matrix calculations simpler.

4 Experimental results

In this section, we present experimental results obtained on the test sets provided for intermediate evaluation in the CLEF ProtestNews track. The ProtestNews evaluation process was divided into two phases. The first phase (intermediate evaluation) was conducted on test sets extracted from Indian news articles. The second phase (final evaluation) includes English-language news articles from both India and China (see below).

The training sets used for experimental testing of the different models are taken from the final evaluation phase (3429 documents and 5884 sentences extracted from India news articles). The test sets are from the intermediate phase and are composed of 457 documents and 663 sentences respectively, extracted from India news articles. Evaluation measures used for each task are precision, recall, F1-score and the average of F1-scores obtained in these two tasks (Avg.2).

Table 1. Experimental results using intermediate evaluation data from CLEF 2019 ProtestNews Track.

Model	Document			Sentence			Avg. 2
	precision	recall	F1-score	precision	recall	F1-score	
xgboost_fast_SVD	0.533	0.392	0.452	0.344	0.072	0.120	0.256
xgboost_fast_wiki_SVD	0.315	0.225	0.263	0.188	0.065	0.097	0.18
xgboost_fast	0.773	0.568	0.655	0.152	0.021	0.037	0.346
xgboost_fast_wiki	0.822	0.637	0.718	0.750	0.391	**0.514**	0.616
svm_fast	0.794	0.529	0.635	0.176	0.108	0.058	0.346
svm_fast_wiki	0.829	0.716	**0.768**	0.833	0.326	0.468	**0.618**
w2v_glove	0.857	0.588	0.698	0.761	0.370	0.498	0.598
w2v_xgboost	0.798	0.618	0.696	0.333	0.007	0.014	0.355
sum_ner	0.535	0.147	0.230	0.461	0.043	0.079	0.155

Table 1 gives results obtained for the model architectures presented in Section 3. At the document level, we observe that the best precision is obtained by the w2v_glove model. For recall and F1-score, best performances are observed with the svm_fast_wiki model. At the sentence level, the best precision is given by svm_fast_wiki. xgboost_fast_wiki obtains the best results both on recall and F1-score. The best average of F1-scores (Avg.2) is given by the svm_fast_wiki model.

It is interesting to note that models based on pre-trained word vectors (i.e. x_wiki, w2v_glove) obtain higher performances than those built directly from the sources provided in the ProtestNews data sets. This finding suggests that, in this context, enriched word vector representation using external data improves global performance of the classification models. The combination of pre-trained word vectors with FastText and a SVM classifier provided a model that was robust and suitable for both levels of scope, especially in terms of precision.

5 Official CLEF ProtestNews results

In this section, we present results obtained for the final evaluation phase of CLEF ProtestNews 2019. Proposed models were evaluated on Task 1 (news article classification) and Task 2 (event sentence detection) using two different testing sets. Task 3 was not attempted.

As in the experimental results above, the training sets provided were composed of 3429 documents (Task 1) and 5884 sentences (Task 2) extracted from English-language news articles from India. Table 2 gives details of the final evaluation test sets.

Table 2. Description of final evaluation test sets.

Test set	number of records	source
task1_test	687	India
china_test_task1	1801	China
task2_test	1107	India
china_test_task2	1235	China

Table 3 gives results for the final evaluation phase. Evaluation measures used are the F1-score for each task and the average of F1-scores obtained across both tasks (Avg. 2). The models presented for this phase are based on the xgboost_fast_SVD and xgboost_fast_wiki_SVD model architectures introduced in Section 3.[6] The best model performance is given for comparison.

The best model achieved an average F1-score across the two tasks of 0.652. As we can observe, the results obtained by our proposed models are less effective, with respectively an average F1-score over the two tasks of 0.163 and 0.193. We

[6] Due to problems with the ProtestNews submission system, only these two models were entered into the final evaluation, despite their relative poor performance in experimental testing.

Table 3. Official results - Final evaluation phase CLEF 2019 ProtestNews Track.

Model	Task 1	Task 2	Avg. 2
Best_run_2019	0.746	0.558	0.652
xgboost_fast_SVD	0.232	0.094	0.163
xgboost_fast_wiki_SVD	0.294	0.092	0.193

suspect that dimension reduction did not offer an advantage here. Usually used on a large set of features, SVD appears not to have helped extraction of suitable discriminant features for these classification tasks. With these settings, results are better on the Task 1 than Task 2. This may be explained by the difference in length of the text records in these two levels of scope. However, it is interesting to observe that the use of a pre-trained model improved results obtained in Task 1.

Comparing performance between experimental testing and the final ProtestNews evaluation, we see a worse Avg. 2 score for xgboost_fast_SVD and slightly better Avg.2 score for xgboost_fast_wiki_SVD, for the final evaluation relative to the experimental test on intermediate evaluation data. We note that in the intermediate phase, models are tested and trained on the same kinds of content (Indian news), whereas in the final phase models are trained on Indian content and tested on both Indian and China content. It appears that use of a pre-trained model is less effective in the sentence level than in the document level when models are applied on the same kind of content. Conversely models trained on similar content are more suitable. We conclude that with these settings, features extracted are less generalisable, while those extracted from a pre-trained model give a slight decrease in performance but are more robust when confronted with another type of data.

6 Conclusion

In this paper, we presented our contribution to the CLEF 2019 ProtestNews Track. Models evaluated combined word-embedding techniques (Word2Vec, GloVe and FastText) with linear classifiers (SVM and XGBoost), as well as dimension reduction as a pre-processing step (SVD). Models showed worse performance when combined with dimension reduction. Word embedding, which is often sensitive to the domain of application, provided best performance when word vectors were generated from pre-trained models, independent of the level of scope.

In future work, we plan to evaluate all models proposed during the experimental phase on the datasets used in the final evaluation phase of CLEF ProtestNews. This will help explore the portability of these models to datasets extracted from an another country and estimate their ability to adapt to new domains.

References

1. Rafeeque, P. C., & S. Sendhilkumar.: A survey on short text analysis in web. In: Third International Conference on Advanced Computing. IEEE, (2011).

2. Kowsari, K., Meimandi, K. J., Heidarysafa, M., Mendu, S., Barnes, L. E., Brown, D. E.: Text Classification Algorithms: A Survey. In: arXiv preprint arXiv:1904.08067. (2019)
3. Belinkov, Y., & Glass, J.: Analysis methods in neural language processing: A survey. Transactions of the Association for Computational Linguistics, vol. 7, pp. 49–72. (2019).
4. Allahyari, M., Pouriyeh, S., Assefi, M., Safaei, S., Trippe, E. D., Gutierrez, J. B., & Kochut, K.: A brief survey of text mining: Classification, clustering and extraction techniques. arXiv preprint arXiv:1707.02919. (2017).
5. Song, G., Ye, Y., Du, X., Huang, X., & Bie, S.: Short text classification: A survey. In: Journal of multimedia, vol. 9 no. 5, pp. 635. (2014).
6. Wang, P., Xu, J., Xu, B., Liu, C., Zhang, H., Wang, F., & Hao, H.: Semantic clustering and convolutional neural network for short text categorization. In: the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing. vol. 2, pp. 352–357. (2015).
7. Young, T., Hazarika, D., Poria, S., & Cambria, E.: Recent trends in deep learning based natural language processing. In: IEEE Computational intelligenCe magazine, vol. 13 n. 3, pp. 55–75. (2018).
8. Joulin, A., Grave, E., Bojanowski, P., & Mikolov, T.: Bag of tricks for efficient text classification. In: arXiv preprint arXiv:1607.01759. (2016).
9. Levy, O., Goldberg, Y., & Dagan, I.: Improving distributional similarity with lessons learned from word embeddings. In: Transactions of the Association for Computational Linguistics, vol. 3, pp. 211–225. (2015).
10. Turian, J., Ratinov, L., & Bengio, Y.: Word representations: a simple and general method for semi-supervised learning. In: the 48th annual meeting of the association for computational linguistics. Association for Computational Linguistics. pp. 384–394. (2010).
11. Ghannay, S., Favre, B., Esteve, Y., & Camelin, N.: Word embedding evaluation and combination. In: the 10th edition of the Language Resources and Evaluation Conference. pp. 300–305. (2016).
12. Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T.: Enriching word vectors with subword information. In: Transactions of the Association for Computational Linguistics, vol. 5, pp. 135–146. (2017).
13. Pennington, J., Socher, R., & Manning, C.: Glove: Global vectors for word representation. In: the 2014 conference on empirical methods in natural language processing (EMNLP). pp. 1532–1543. (2014).
14. Mikolov, T., Chen, K., Corrado, G., & Dean, J.: Efficient estimation of word representations in vector space. In: arXiv preprint arXiv:1301.3781. (2013).
15. Wang, S., & Manning, C. D.: Baselines and bigrams: Simple, good sentiment and topic classification. In: the 50th annual meeting of the association for computational linguistics: vol. 2. pp. 90–94. Association for Computational Linguistics. (2012).