Classification Of Spanish Election Tweets (COSET) 2017 : Classifying Tweets using Character and Word Level Features

by

Ankush Khandelwal, Sahil Swami, Syed S. Akhtar, Manish Shrivastava

in

Second Workshop on Evaluation of Human Language Technologies for Iberian Languages
(*IBEREVAL-2017*)

Murcia, Spain

Report No: IIIT/TR/2017/-1

Centre for Language Technologies Research Centre
International Institute of Information Technology
Hyderabad - 500 032, INDIA
September 2017

Classification Of Spanish Election Tweets (COSET) 2017 : Classifying Tweets using Character and Word Level Features

Ankush Khandelwal[1] Sahil Swami[2] Syed S. Akhtar[3] M. Shrivastava[4]

International Institute of Information Technology
Hyderabad, Telangana, India
{ankush.k,sahil.swami,syed.akthar}@research.iiit.ac.in,
manish.shrivastava@iiit.ac.in

Abstract. This paper describes the International Institute of Information Technology of Hyderabad's submission to the task Classification Of Spanish Election Tweets (COSET) as a part of IBEREVAL-2017[1]. The task is to classify Spanish election tweets into political, policy, personal, campaign and other issues. Our system uses Support Vector Machines with radial basis function kernel to classify tweets. We dwell upon the character and word level features along with the word embeddings and train the classification model with them and present the results. Our best run achieves a F_1-macro score of 0.6054 on the test corpus for first phase and 0.8509 for the second phase.

Keywords: SVM, Random forest, Decision tree, Extra tree, Twitter, Classification, Machine Learning, Word2vec, Radial basis function kernel.

1 Introduction

Classification of natural language texts is one of the classic challenges in natural language processing. Different platforms such as blogs, social networks, microblogs provide indispensable amount of information which is valuable for academic as well as for commercial purposes.

Tweet classification is the task to automatically classify a tweet into one of the predefined classes. This paper analyzes tweets which talks about 2015 Spanish General Election and classifies them into five categories, namely, political issues which are related to the most abstract electoral confrontation, policy issues which describes sectoral policies, personal issues which talks about the life and activities of the candidates, campaign issues which are related with the evolution of the campaign and other issues using supervised learning techniques. For example, consider the tweet *"PSOE leader @sanchezcastejon is talking. He is also "immensely proud". Supporters cheering "presidente, presidente" https://t.co/Wu2IgxnIr1 "*. This tweet belongs to political issues based on the description of classes.

We develop an automated system for classifying election tweets in a set of classes using character and word level features combined with pre-trained set of Spanish word embeddings[2]. We use character and word vectors as features by adopting bag of words approach and experiment with three classifiers, kernel Support Vector Machines (SVM), random forests and extra tree classifier. In this paper, we present a system classifying tweets which uses character and word level features and SVM with radial basis function kernel for classification.

Previous research in classification of tweets includes supervised polarity classification of tweets [3]. They have used a hybrid approach combining machine learning and natural language processing knowledge for identifying the polarity in tweets. They classified tweets into six classes determining the opinion in the tweets. [4] adopt a graph based approach to classify tweets in a predefined set of topics and attain 70% accuracy. [5] uses SVMs for text classification.

The structure of the paper is as follows, we begin by describing the corpus in Section 2, then we explain our system's architecture in Section 3 which talks about corpus pre-processing followed by feature extraction. In the next subsection, we describe the classification models and the results of the experiments conducted using character and word level features. In the last section, we conclude the paper followed by the scope of improvements and the bibliography.

2 Dataset

We use the corpus provided by the organizers of *COSET 2017*[1]. In the corpus for the first phase, training data consists of 2242 tweets of Spanish General Election 2015 , 251 for development and 624 for testing and their distribution in five categories is shown in table 1. Each of the tweets in training and development is annotated with one of the five categories. The second phase consists of classifying approximately 16 million tweets using the best classification model from the first phase.

Table 1. The distribution of classes in training and development corpus

corpus	Political	Policy	Personal	Campaign	Other
Training	530	786	511	152	263
Dev	57	88	71	9	25

3 System Architecture

3.1 Pre-processing

First step consists of tokenization in which all the words in a tweet are separated using space as the delimiter and then converted to lower cases, following by the removal of punctuation marks. The words starting with @ (mentions) and # (hashtags) are kept same. After storing all the hashtags and mentions, we removed '#' symbol from all the hashtags and the word is decomposed using camel cases and underscore (_), as most of the hashtags in training and test corpus either comprise of camel case format or combined with underscore. We segregated such words and added them to the tokenized tweet (Hashtag decomposition)[6]. For example, *#CarlosBarraGalán* is decomposed to Carlos, Barra and Galan. If there are some outliers then we adopt the approach of recursively finding the words in the tweet [6]. Mentions and urls are converted to "MENTION" and "URL" and are stored in the tokenized tweet. In the next step, the Spanish stopwords are removed from the tweets and the tokens are reduced to the root form using Snowball [10] stemmer implemented in NLTK. Finally, tokenized tweets are stored along with their respective classes.

3.2 Features

This subsection describe the features that we have used in our systems to build attribute vectors for training our classification models[1].

Character N-grams Since the number of all existing n grams is very large, we downsample them using their frequency. We have taken only those n grams which occur at least ten times in the training corpus which reduces the size of the feature vector. The main advantage of this feature is that it is language independent and does not require any previous knowledge or pre-processing steps like tokenization, stemming and stop words removal.

Word N-grams We also take into account the word n grams where n varies from 1 to 3. Word n-grams have proven to be important features for text classification in previous researches [13]. In this case we take only those n-grams which occur at least ten time in the corpus.

Reference tokens for each class We identified the tokens which occurs for more than 60% in a class and occur more than five times in the corpus set and took them as a feature for the classification models [12]. We calculated the score for each token as

$$Score(token) = max_{class\_label \in class1/2/3/4/5} \frac{freq(token, class\_label)}{freq(token)}$$

Only those tokens are taken as features for classification which have a score ≥ 0.6 and occur at least five times in the training corpus.

[1] All the thresholds mentioned have been decided after empirical fine tuning.

3.3 Word Embeddings

Next step of building feature vector involves augmenting the feature vector with pre-trained word2vec embeddings [2] of Spanish words [5]. There have been tremendous use of word embeddings in text classification [7][8]. To represent a tweet using the word embeddings provides by Word2vec model, we extract the embeddings for all the words in a tweet and take the mean of all the embeddings of the words in the tweet[9]. For example, if a tweet contains 10 words then we extract the embeddings of each of the ten words and take the average of all the ten embeddings. Then we append this representation to our feature vector to form the final training samples.

3.4 Classification approach

We experiment with three different techniques for classification: Support Vector Machines (SVM), Random Forests and Extra Tree classifier. For training our classifiers on labelled tweets we have used python library Scikit-learn [11]. Since our representation produces a long feature vector, Support Vector Machine is used with rbf kernel as they are efficient with high dimensional corpus because their ability to learn is based on the margin with which they separate the corpus and independent of the dimensionality of feature space. Moreover, support vector machines have proven to be an efficient model for text classification and twitter sentiment analysis *(Pilaszy, 2005)*.

Random forests and extra tree classifiers are used because they are efficient with numerical feature vectors and their ability to reduce over fitting by training on feature subspaces. We perform grid search on every classifier for tuning the parameters.

3.5 Results and Evaluation

Table 2 shows the results of the experiments conducted on development corpus for parameter tuning with different classification models :

Table 2. Feature-wise accuracy (in %) for classification of Spanish tweets.

Feature	Kernel SVM	Random Forest	Extra Tree
Character N-grams	60.8	61.2	60
Word N-grams	52.8	50.8	47.6
Reference tokens	63.6	60.4	58.4
Reference hashtags	39.2	42	42.4
Mentions	34.8	38	38.4
All features	69.8	67	68

For parameter tuning, maximum accuracy of 70% on development corpus is achieved by training kernel svm with character n-grams and reference tokens with word embeddings. For the evaluation of the systems, F_1-macro metric is used which calculates the score as follows:

$$F_{1-macro} = \frac{1}{|L|} \sum_{l \in L} F_1(y_l, \hat{y}_l)$$

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

$$precision = \frac{1}{|L|} \sum_{l \in L} Pr(y_l, \hat{y}_l)$$

$$recall = \frac{1}{|L|} \sum_{l \in L} R(y_l, \hat{y}_l)$$

Fig. 1. F_1-macro Metric

Our best system was able to achieve a F_1-macro score of 0.60 while the best performing model has 0.64. In addition, our svm model achieves a F_1-macro score of 0.8506 on the corpus for second phase which consists of 16 million tweets which is further used by the COSET [1] for the making a large labelled corpus.

Kernel svm with character and word level features along with word embeddings achieves the best F_1-macro score amongst other runs submitted. Random forest classifier performs close to the baseline model having a score of 0.4435 only. Since the amount of training corpus available is small, random forest did not perform well on unseen corpus as it overfits the development corpus.

4 Conclusion and Future Work

In this task, we classify Spanish election tweets into a set of predefined classes using supervised classifiers by taking character and word level features. An important point to note is that character level features performs better than the word level for tweet classification in all our experiments. Taking words in the hashtag and augmenting word embeddings to the feature vector improves the accuracy of classification on development corpus. Our best model uses support vector machines with radial basis functions and achieve F_1-macro score of 0.6054 on test data for the first phase and 0.8509 for the second phase. Future work includes training the classifiers by incorporating a large corpus and taking features like POS tagging, NER etc. Moreover, several other supervised and unsupervised machine learning algorithms can used for classification.

References

[1] Giménez M., Baviera T., Llorca G., Gámir J., Calvo D., Rosso P., Rangel F. Overview of the 1st Classification of Spanish Election Tweets Task at IberEval 2017. In: Notebook Papers of 2nd SEPLN Workshop on Evaluation of Human Language Technologies for Iberian Languages (IBEREVAL), Murcia, Spain, September 19, CEUR Workshop Proceedings. CEUR-WS.org, 2017.

[2] Cristian Cardellino: Spanish Billion Words Corpus and Embeddings (March 2016), *http://crscardellino.me/SBWCE/*

[3] Vilares, David, Miguel A. Alonso, and Carlos Gómez-Rodríguez. "Supervised Polarity Classification of Spanish Tweets based on Linguistic Knowledge."

[4] Cordobés, Héctor, et al. "Graph-based Techniques for Topic Classification of Tweets in Spanish." *IJIMAI 2.5 (2014): 32-38.*

[5] Joachims T. (1998) Text categorization with Support Vector Machines: Learning with many relevant features. In: Nédellec C., Rouveirol C. (eds) Machine Learning: ECML-98. ECML 1998. Lecture Notes in Computer Science (Lecture Notes in Artificial Intelligence), vol 1398. Springer, Berlin, Heidelberg

[6] Billal, Belainine, Alexsandro Fonseca, and Fatiha Sadat. "Named Entity Recognition and Hashtag Decomposition to Improve the Classification of Tweets." *WNUT 2016 (2016): 102.*

[7] Dasgupta, Surajit, et al. "Word Embeddings for Information Extraction from Tweets."

[8] Yang, Xiao, Craig Macdonald, and Iadh Ounis. "Using word embeddings in twitter election classification." arXiv preprint arXiv:1606.07006 (2016).

[9] Kenter, Tom, Alexey Borisov, and Maarten de Rijke. "Siamese cbow: Optimizing word embeddings for sentence representations." *arXiv preprint arXiv:1606.04640 (2016).*

[10] Porter, Martin F. "Snowball: A language for stemming algorithms."(2001).

[11] Buitinck, Lars, et al. "API design for machine learning software: experiences from the scikit-learn project." arXiv preprint arXiv:1309.0238 (2013).

[12] Mohammad, Saif M., Parinaz Sobhani, and Svetlana Kiritchenko. "Stance and sentiment in tweets." arXiv preprint arXiv:1605.01655 (2016).

[13] Cavnar, William B., and John M. Trenkle. "N-gram-based text categorization." Ann arbor mi 48113.2 (1994): 161-175.