

# LABDA's early steps toward Multimodal Stance detection

Isabel Segura-Bedmar<sup>[0000-0002-7810-2360]</sup>

Universidad Carlos III de Madrid, Leganés 28911, Madrid, Spain  
[isegura@inf.uc3m.es](mailto:isegura@inf.uc3m.es)  
<http://labda.inf.uc3m.es>

**Abstract.** In this paper, we describe our participation at the task on MultiModal Stance Detection in Tweets on Catalan 1Oct Referendum. Tweets are cleaned and represented using the simple Bag-of-Words approach with tf-idf vectors. Then, we explore the most widely used and efficient classifiers in text classification. Some algorithms are adapted to be multi-class learning by using one-versus-all strategy because they are naturally binary. For each algorithm, we perform grid search on all combinations of its parameters in order to find the set of parameters which provides the most accurate model. Our system employing text and context obtains the top macro F1 (28.02%) for spanish tweets.

**Keywords:** Multimodal stance detection · Machine Learning.

## 1 Introduction

The goal of the stance detection is to determine the stance of the author of a text with respect to a specific topic. The stance can take the following values: favor (positive), against (negative) or neutral [5]. In last years, several shared tasks on stance detection have been organized such as SemEval-2016 Task 6: Detecting Stance in Tweets [5] and Stance and Gender detection in tweets on Catalan Independence (StanceCat 2017) [7]. In 2018, a new shared task, MultiStanceCat, is organized with the goal of detecting the author's stance about the Catalan independence referendum, which was hold on 1 October 2017. The MultiStanceCat 2018 task [8] goes one step further than these previous shared tasks and does not only provide the texts of the tweets, but also gives their previous and next tweets and the images from the authors timeline. Thus, the participating systems can develop approaches that exploit text and images to infer the stance expressed in the tweets.

## 2 The system

For lack of time and experience on visual computing, we decided to use an approach that only exploits the text of tweets. The stance detection can be

formulated as a multi-class problem with three classes (*FAVOR*, *AGAINST* and *NEUTRAL*).

We performed an exhaustive evaluation of the most widely used and efficient classifiers in text classification. In particular, we used the following algorithms: Multinomial Naive Bayes [4], Linear Support Vector Machine (SVM)[9], Logistic Regression [10], k-Nearest Neighbours algorithm (k-NN) [1], Decision Trees [6] and Random Forest [2]. As some of these classifiers are binary (Linear SVM, Logistic Regression and Multinomial NB), they must be adapted to a multi-class classification problem by using the one-versus-all strategy. All the experiments were conducted in Python using Scikit Learn for classification.

Multinomial Naive Bayes classifier has been proven very effective for text classification. It is a probabilistic model based on theorem of Bayes. This classifier calculates the probabilities of each text belonging to each class and then selects the class with the maximum probability. The adjective naive comes from the assumption that all features are independent given class. Although such an independence assumption is not usually true, the algorithm often performs surprisingly well with a fast computational time. Moreover, it requires a small amount of training data, is very easy to implement and is also very scalable. Despite its simplicity, the Naive Bayesian classifier often exceeds more sophisticated classification algorithms.

SVM, perhaps one of the most popular and successful classifiers, is a non-probabilistic linear classifier that tries to find the hyperplane that best separates the classes, maximizing the margin between them while, at the same time, minimizing the number of misclassification errors. The main reason of its success is that most text classification problems are linearly separable [3]. Moreover, SVM is able to learn, irrespective of the dimensionality of the feature space, because it is based on maximization of the margin, not the number of features [3]. If the classes are separable by a wide margin, then the model will be able to generalize even with a very large number of features. There are several kernel functions such as linear kernel, polynomial kernel, sigmoid kernel or radial basis function (RBF) kernel. A kernel function transforms the input space into a high dimensional space where the problem can be represented as a linear problem. Linear kernel is much faster, while RBF generally provides better performance. However, when the number of features is large, which is typical in text classification, the RBF kernel does not provide better performance than using the linear kernel. In our experiments, we only tried with linear kernel.

Logistic Regression is a linear classifier, which can be used to predict the probability of an event. Its main advantage is that its results have an easier interpretation than those obtained by other classification algorithms. Moreover, this algorithm provides a regularization parameter to avoid over-fitting. Among their disadvantages, it requires much more data than other classifiers to obtain stable and accurate results. Moreover, it is not able to capture complex relationships in the data.

k-NN is one of the simplest classification algorithm. It is based on the idea that the closer instances are, the more probability they belong to the same class.

In this way, one of its main advantages is that it is a lazy classifier because it does not create a training model from the training dataset, but rather compares the test instance with all instances to determine its class. Moreover, the classifier does not depend on the data distribution.

Random forest is an ensemble classifier of a collection of decision trees by randomly selecting examples from the training data. The final prediction is calculated by aggregating the predictions of each tree. Learning from different trees leads to mitigate the over-fitting as well as errors due to bias and variance in the decision trees. Random forests are more robust and generally exhibit better results than decision trees.

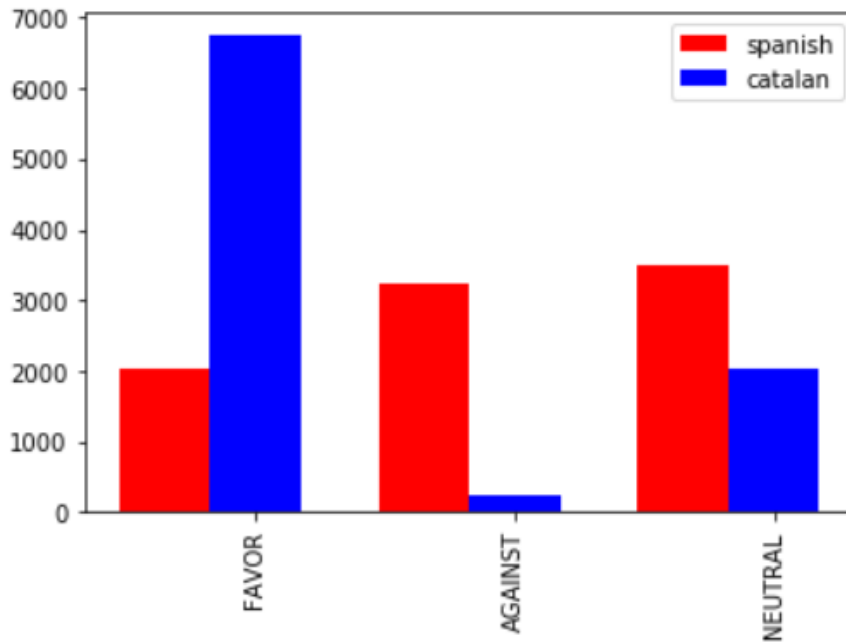


Fig. 1. Stance distribution of the training dataset.

Figure 1 shows the distribution of the tweets labelled with stance in the training dataset. Most of the tweets written in Catalan are clearly in favor of holding the referendum. However, for the tweets written in Spanish, the stance seems to be distributed more-or-less equally in the three classes. *AGAINST* and *NEUTRAL* have a very close number of tweets (above 3,200), while *FAVOR* is the class with less tweets (around 2,000). This more balanced distribution may help the learning of the algorithms for Spanish tweets, while the task may be more difficult for tweets written in Catalan because the classes are not balanced

(there are very few instances for the *AGAINST* class). The training dataset contains a total of 8,764 tweets written in Spanish and 9,009 written in Catalan.

We performed some experiments in order to determine if the context tweets could help in the task. The results were positive, and thereby, we decided to include the previous and next tweets of each tweet as part of it. We also tried with the StanceCat 2017 dataset, however in this case, the experiments showed that it did not improve the performance. Thus, finally, the StanceCat 2017 dataset was not used for training our system. The tweets were represented using the simple Bag-Of-Word (BoW) approach, but instead of using the word frequencies, we used their inverse document frequencies (tf-idf) to measure the word relevance in the whole collection of tweets. To do this, we used the TfidfVectorizer class to convert the tweets into tf-idf values.

As the organizers have not provided any validation set, we randomly generate a test dataset (20% from the training dataset). To do this, we used the StratifiedShuffleSplit class that provides a random split with same balance of classes. Moreover, we performed grid search on all combinations of the parameters and for each classifier in order to find best setting (see Table 1). We used the GridSearchCV class.

**Table 1.** Best settings.

Algorithm	Parameters
Linear SVM	C=1, class_weight=balanced
Logistic Regression	C=1, class_weight=balanced
MultinomialNB	alpha=0.01
kNN	n_neighbors=3, weights=distance
Decision Tree	max_depth=14
Random Forest	n_estimators=700

### 3 Results

Table 2 shows the results obtained by the different classifiers (with the best settings) on the validation dataset of tweets written in Spanish. Linear SVM is the algorithm that obtains best performance for classes *FAVOR* and *AGAINST*, while the top F1 for the other class, *NEUTRAL*, is provided by Random Forest (F1=76%). The second top F1 for *NEUTRAL* is achieved by Linear SVM. Therefore, we concluded, based on these results, that Linear SVM was the best classifier for stance detection in Spanish tweets.

For the tweets written in Catalan (see Table 3), we observed that *FAVOR* obtained much better results than the other classes (F1=91%). This may be due to the number of its instances in the training dataset, which is significantly greater than the other classes. Similarly, the classifiers also provided much worse performance for *AGAINST*, because there are very few instances in the training

**Table 2.** Results for Spanish tweets on the validation set.

Algorithm	Class	P	R	F1
Linear SVM	FAVOR	0.77	0.70	<b>0.73</b>
	AGAINST	0.69	0.76	<b>0.72</b>
	NEUTRAL	0.74	0.76	0.75
	Average	0.74	0.73	0.73
LogisticRegression	FAVOR	0.75	0.63	0.69
	AGAINST	0.62	0.76	0.68
	NEUTRAL	0.73	0.74	0.74
	Average	0.71	0.71	0.71
Multinomial Naive Bayes	FAVOR	0.69	0.72	0.70
	AGAINST	0.68	0.67	0.67
	NEUTRAL	0.72	0.69	0.70
	Average	0.70	0.70	0.70
kNN	FAVOR	0.67	0.65	0.66
	AGAINST	0.66	0.54	0.59
	NEUTRAL	0.63	0.72	0.67
	Average	0.65	0.65	0.65
Decision Tree	FAVOR	0.67	0.54	0.60
	AGAINST	0.57	0.77	0.71
	NEUTRAL	0.65	0.77	0.71
	Average	0.64	0.64	0.63
Random Forest	FAVOR	0.69	0.70	0.69
	AGAINST	0.74	0.56	0.64
	NEUTRAL	0.72	0.80	<b>0.76</b>
	Average	0.71	0.71	0.71

dataset. Linear SVM obtained the top F1 for the classes *FAVOR* (F1=91%) and *NEUTRAL* (F1=75%). Random Forest and Multinomial NB also achieved the top F1 for *FAVOR*.

Based on the experiment results, we decided to use Linear SVM to process the test dataset.

We sent two different runs: using the context tweets and without using them. The organizers published the final results and our classifier using text and context information achieved the top macro F1 (0.2802) for Spanish. However, this setting is the fourth place for Catalan with a macro F1 of 0.2876 (the top F1 was 0.3068).

## 4 Conclusion

Our system is a very simple approach, which only exploits the tweets. The task is very attractive and there is much room for improvement. We will try machine learning classifiers trained using hand-engineered features as well as word embeddings. We also plan to extend our research by using deep learning methods.

**Table 3.** Results for Catalan tweets on the validation set.

Algorithm	Class	P	R	F1
Linear SVM	FAVOR	0.92	0.91	<b>0.91</b>
	AGAINST	0.76	0.32	0.45
	NEUTRAL	0.72	0.79	<b>0.75</b>
	Average	0.87	0.79	0.86
LogisticRegression	FAVOR	0.93	0.87	0.90
	AGAINST	0.32	0.40	0.36
	NEUTRAL	0.67	0.78	0.72
	Average	0.85	0.84	0.85
Multinomial Naive Bayes	FAVOR	0.80	0.95	<b>0.91</b>
	AGAINST	0.81	0.34	<b>0.48</b>
	NEUTRAL	0.80	0.61	0.69
	Average	0.85	0.86	0.85
kNN	FAVOR	0.87	0.92	0.90
	AGAINST	0.56	0.28	0.37
	NEUTRAL	0.69	0.61	0.65
	Average	0.82	0.83	0.82
Decision Tree	FAVOR	0.80	0.96	0.88
	AGAINST	0.60	0.06	0.11
	NEUTRAL	0.71	0.30	0.43
	Average	0.77	0.79	0.75
Random Forest	FAVOR	0.84	0.98	<b>0.91</b>
	AGAINST	1	0.08	0.15
	NEUTRAL	0.85	0.50	0.63
	Average	0.85	0.84	0.82

## Acknowledgments

This work was supported by the Research Program of the Ministry of Economy and Competitiveness - Government of Spain, (DeepEMR project TIN2017-87548-C2-1-R).

## References

1. Altman, N.S.: An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician* **46**(3), 175–185 (1992)
2. Breiman, L.: Random forests. *Machine Learning* **45**(1), 5–32 (2001)
3. Joachims, T.: Text categorization with support vector machines: Learning with many relevant features. *Machine Learning* pp. 137–142 (1998)
4. Langley, P., Iba, W., Thompson, K.: An analysis of bayesian classifiers. In: *AAAI*. vol. 90, pp. 223–228 (1992)
5. Mohammad, S., Kiritchenko, S., Sobhani, P., Zhu, X., Cherry, C.: Semeval-2016 task 6: Detecting stance in tweets. In: *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*. pp. 31–41 (2016)
6. Quinlan, J.R.: Induction of decision trees. *Machine Learning* **1**(1), 81–106 (1986)

7. Taulé, M., Martí, M.A., Rangel, F.M., Rosso, P., Bosco, C., Patti, V.: Overview of the task on stance and gender detection in tweets on catalan independence at ibereval 2017. In: 2nd Workshop on Evaluation of Human Language Technologies for Iberian Languages, IberEval 2017. vol. 1881, pp. 157–177. CEUR-WS (2017)
8. Taulé, M., Rangel, F., Martí, M.A., Rosso, P.: Overview of the task on multimodal stance detection in tweets on catalan 1oct referendum. In: Proceedings of the Third Workshop on Evaluation of Human Language Technologies for Iberian Languages (IberEval 2018), Seville, Spain (2018)
9. Vapnik, V.: The nature of statistical learning theory. Springer Science & Business Media (1995)
10. Walker, S.H., Duncan, D.B.: Estimation of the probability of an event as a function of several independent variables. *Biometrika* **54**(1-2), 167–179 (1967)