TASS 2015, septiembre 2015, pp 71-74 recibido 20-07-15 revisado 24-07-15 aceptado 29-07-15

BittenPotato: Tweet sentiment analysis by combining multiple classifiers.

BittenPotato: Análisis de sentimientos de tweets mediante la combinación de varios clasificadores.

Iosu Mendizabal Borda
(IIIA) Artificial Intelligence Research Institute
(CSIC) Spanish Council for Scientific Research
iosu@iiia.csic.es

Jeroni Carandell Saladich
(UPC) Universitat Politecnica de Catalunya
(URV) Universitat Rovira i Virgili
(UB) Universitat de Barcelona
jeroni.carandell@gmail.com

Resumen: En este artículo, usamos un saco de palabras (bag of words) sobre n-gramas para crear un diccionario de los atributos más usados en una dataset. Seguidamente, aplicamos cuatro distintos clasificadores, el resultado de los cuales, mediante diversas técnicas pretende mostrar la polaridad real de la frase extrayendo el sentimiento que contiene.
Palabras clave: Análisis de Sentimientos, Procesamiento de lenguaje natural.

Abstract: In this paper, we use a bag-of-words of n-grams to capture a dictionary containing the most used "words" which we will use as features. We then proceed to classify using four different classifiers and combine their results by apply a voting, a weighted voting and a classifier to obtain the real polarity of a phrase.
Keywords: Tweet sentiment analysis, natural language processing.

1 Introduction and objectives

Sentiment analysis is the branch of natural language processing which is used to determine the subjective polarity of a text. This has many applications ranging from the popularity of a certain product, the general opinion about an event or politician among many others.

In the particular case of twitter texts, these have the misfortune or great advantage of only consisting of a maximum of 140 characters. The disadvantage is that short texts aren't very accurately describable with bag of words which we will use, on the other hand, the limit also forces the author of the tweet to be concise in its opinion and therefore noise or non relevant statements are usually left out.

In this workshop for sentiment analysis focused on Spanish, a data set with tagged tweets according to their sentiment is given along with a description of evaluation measures as well as descriptions of the different tasks (Villena-Román et al., 2015).

The rest of the article is laid out as follows: Section 2 introduces the architecture and components of the system, namely the pre-processing, the extraction of features, the algorithms used and then the process applied to their results to obtain our final tag. Section 3 analyses the results obtained in this workshop. Finally, to conclude, in section 4 we will draw some conclusions and propose some future work.

2 Architecture and components of the system

Our system contains four main phases: data pre-processing, feature extraction - vectorization, the use of classifiers from which we extract a new set of features and finally a combined classifier which uses the latter to predict the polarity of the text.

2.1 Pre-processing

This step, crucial to all natural language processing task, consists of extracting noise from the text. Many of the steps such as removal of URLs, emails, punctuation, emoticons, spaced words etc. are general and we will not get into so much, yet some are more particular to the language in particular, such

Publicado en http://ceur-ws.org/Vol-1397/. CEUR-WS.org es una publicación en serie con ISSN reconocido ISSN 1613-0073

as the removal of letters that are repeated more than twice in Spanish.

2.2 Vectorization: Bag of words

In order to be able to apply a classifier, we need to turn each tweet into a vector with the same features. To do this, one of the most common approach is to use the Bag-of-Words model with which given a corpus of documents, it finds the N most relevant words (or n-grams in our case). Each feature, therefore represents the appearance of a different relevant "word". Although the relevance of a word can be defined as the number of times it appears in the text, this has the disadvantage of considering words that appear largely throughout the whole document and lack semantic relevance. In order to counter this effect a more sophisticated approach called tf-idf (term frequency - inverse term frequency) is used. In our project we used the Scikit-Learn TfidfVectorizer (Pedregosa et al., 2011) to convert each tweet to a length N feature vector.

2.3 Algorithms: Classifiers

Once we have a way of converting sentences into a representation with the same features, we can use any algorithm for classification. Again, for all of the following algorithms we used the implementations in the Scikit-Learn python package (Pedregosa et al., 2011).

2.3.1 SVM

The first simple method we use is a support vector machine with a linear kernel in order to classify. This is generally the most promising in terms of all the used measures both with the complete of reduced number of classes.

2.3.2 AdaBoost (ADA)

Adaboost is also a simple, easy to train since the only relevant parameter is the number of rounds, and it has a strong theoretical basis in assuring that the training error will be reduced. However, this is only true with enough data (Freund and Schapire, 1999), given that the large number of features (5000) compared to the number of instances to train (around 4000 because of the cross validation with the training data that we will use to test the data), this is the worst performing method as can be seen in tables 1 and 2.

2.3.3 Random Forest (RF)

We decided to use this ensemble method as well because it has had very positive effects with accuracies that at times surpass the AdaBoosts thanks to its robustness to noise and outliers (Breiman, 2001).

2.3.4 Linear Regression (LR)

Since the degrees of sentiment polarity are ordered, we decided that it would also be appropriate to consider the problem as a discrete regression problem. Although a very straightforward approach, it seems to give the second best results in general at times surpassing the SVM (Tables 1 and 2).

2.4 Result: Combining classifiers

After computing the confusion matrices of the used classifiers we reached the conclusion that certain algorithms were better at capturing some classes than others. These confusion matrices can be observed in the next Section 3. Because of this reason we decided to combine the results of different classifiers to have more accurate results. In other words, we use the results of the single classifiers as a codification of the tweet into lower dimension. We can interpret each single classifier as an expert that gives its diagnose or opinion about the sentiment of a tweet. Since these different experts can be mistaken and disagree, we have to find the best result by combining the latter.

We tried three different combining methods. The first method is a simple voting of the different classifiers results and the more repeated one wins, in case of draws a random of the drawing ones will win. The second proposal is a more sophisticated voting with weights in each of the classifier results, these weights are computed with a train set and are normalized accuracies of the classification of this set.

Finally, the third method consists of another classification algorithm, this time of results. The idea is that we treat each previous classifier as an expert that give its own diagnose of the tweet, given that we have the real tweets, we decided to train a Radial Basis Function (RBF) with all of the training dataset and afterwards use the RBF to classify the final test results, which were the results we uploaded to the workshop. All three of these methods enhanced our results by few yet significant points. This can be thought of as a supervised technique for dimensionality

reduction, since we convert a dataset of 5000 features into only 4.

3 Empirical analysis

We are now going to analyse the results obtained in the workshop with the given testing tweet corpus. This section is separated in two subsections, firstly we will introduce the results obtained with the use of the four classifiers explained in Section 2.3. Secondly, we will focus on the usage of the three combining methods introduced in Section 2.4.

3.1 Single classifiers

First of all we are going to talk about the results obtained with the simple use of the four single classifiers explained in Section 2.3. The analysis is done with two different data sets; on the one hand a set separated in four classes and on the other hand a data set separated in six classes.

As it is depicted in Tables 1 and 2 the SVM and the Linear Regression classifiers are the most optimal ones in terms of the f1-measure which is the harmonic mean between the recall and the precision.

	Acc	Precision	Recall	F1
SVM	57.6667	0.4842	0.4759	0.4707
AB.	49.3333	0.4193	0.4142	0.4072
RF	54.0000	0.5122	0.4105	0.3968
LR	59.3333	0.4542	0.4667	0.4516

Table 1: Average measures in 3-Cross validated classifiers for 4 classes.

	Acc	Precision	Recall	F1
SVM	40.3333	0.3587	0.3634	0.3579
AB	35.0	0.3037	0.3070	0.2886
RF	39.3333	0.3370	0.3267	0.2886
LR	42.3333	0.3828	0.3621	0.3393

Table 2: Average measures in 3-Cross validated classifiers for 6 classes.

Observing the confusion matrix of the previously mentioned techniques, Random forest and Linear regression, we can learn perhaps more about the data itself. For instance, that the number of Neutral tweets are so low that tweets are rarely classified as such as seen in the NEU columns of the confusion matrices in figures 2 and 1. Another curious fact is that P+ labels are very separable for our classifiers. This could be because extremes might contain most key words that determine a positive review as opposed to the more subtle class P.

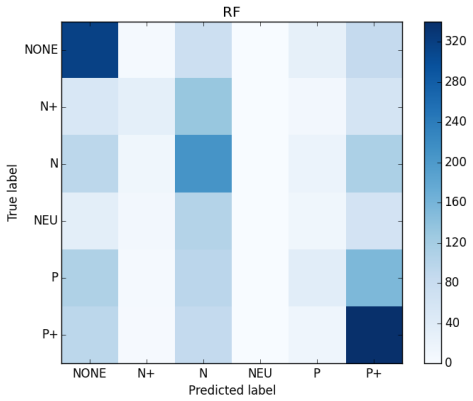

Figure 1: Confusion Matrix for a Random Forest with 6 classes.

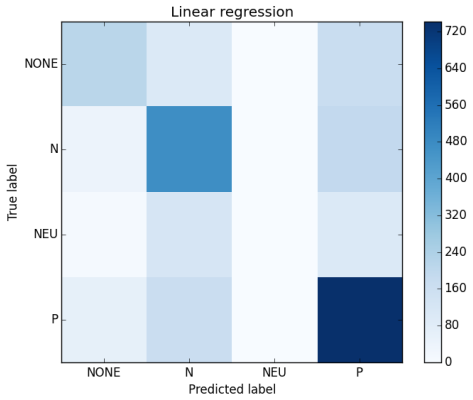

Figure 2: Confusion Matrix for Linear Regression with 4 classes.

3.2 Combining classifiers

After applying the 4 previous single classifiers to each tweet, we obtain a data matrix where each features correspond to the label set by each classifier. We can interpret this as some sort of dimensionality reduction technique where we now have a tweet transformed into an element of 4 attributes each corresponding to a classifier's results.

In tables 3 and 4 we can see the official results of the three combined classifiers on the Train data.

We have to keep in mind that when we are comparing the combined classifiers with the single classifiers, we are using two different

test sets. In the single classifiers, we use 3-Cross Validation exclusively on the train data to obtain average measure for each classifier. With the combined classifiers, we trained on the Train set and evaluated on the final Test set.

Notice that the weighted voting outperforms the normal voting. This seems intuitive because the weighted voting gives more importance to the most reliable classifiers. The RBF's results are not as promising as the previous two methods but it still outperforms all of the single classifiers.

	Acc	Precision	Recall	F1
Voting	59.3	0.500	0.469	0.484
Weighted Voting	59.3	0.508	0.465	0.486
RBF	60.2	0.474	0.471	0.472

Table 3: Official Results for the combined classifiers for 4 classes.

	Acc	Precision	Recall	F1
Voting	53.5	0.396	0.421	0.408
Weighted Voting	53.4	0.402	0.430	0.415
RBF	51.4	0.377	0.393	0.385

Table 4: Official results for the combined classifiers for 6 classes.

In general we can see that these methods, with the exception of the SVM in terms of F1-measure, outperform the rest.

4 Conclusions and future work

In this paper we have described our approach for the SEPLN 2015 for the global level with relatively good results considering the number of classes, and the general difficulty of the problem.

We have started by describing the initial preprocessing and the extraction of features using a bag of words on trigrams and bigrams. Then we have described and compared four different classifiers that we lated used as a way of translating the data into merely 4 dimensions, from 5000.

We can conclude that multiple classifiers are good at capturing different phenomena and that by combining them we tend to have a better global result as we have obtained in most of the TASS 2015 results of the Global level.

In general we are satisfied with the results obtained of the TASS2015 challenge. As future work, we propose to explore different classifiers that might capture different phenomena so that the combined classifier might have more diverse information. Also different combined classifiers should be trained.

References

Breiman, L. 2001. Random forests. *Machine Learning*, 45(1):5–32.

Freund, Y. and R. E. Schapire. 1999. A short introduction to boosting.

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python . *Journal of Machine Learning Research*, 12:2825–2830.

Villena-Román, J., J. García-Morera, M. A. García-Cumbreras, E. Martínez-Cámara, M. T. Martín-Valdivia, and L. A. Ureña López. 2015. Overview of tass 2015.