A Comparison of Algorithms for Text Classification of Albanian News Articles

Arbana Kadriu
SEE University, Macedonia
Lejla Abazi
SEE University, Macedonia

Abstract

Text classification is an essential work in text mining and information retrieval. There are a lot of algorithms developed aiming to classify computational data and most of them are extended to classify textual data. We have used some of these algorithms to train the classifiers with part of our crawled Albanian news articles and classify the other part with the already learned classifiers. The used categories are: latest news, economy, sport, showbiz, technology, culture, and world. First, we remove all stop words from the gained articles and the output of this step is a separate text file for each category. All these files are then split in sentences, and for each sentence the appropriate category is assigned. All these sentences are then projected to a single list of tuples sentence/category. This list is used to train (80% of the overall number) and to test (the remained 20%) different classifiers. This list is at the end shuffled aiming to randomize the sequence of different categories. We have trained and then test our articles measuring the accuracy for each classifier separately. We have also analysed the training and testing time.

Keywords: data mining, text classification, news articles, machine learning
JEL classification: C00, C30

Introduction

In text classification, the question is: if we have a representation of a document d and a fixed set of classes $C = \{c_1, c_2, \ldots, c_n\}$, how to build classification functions ("classifiers"). That is how to determine category of d: $\gamma(d) \in C$, where γ(d) is a classification function (Manning et al., 2008). The classification problem belongs to supervised learning approach, where input data must be a hand-classified data aiming to train a classifier. This way of creating hand-labelling training data, is very time consuming (Jurka et al., 2013). For languages that there is an industry interest in text mining, there are considerable pre-trained classified data. From over 6000 languages from all corners of the world, only a small number (less than 100) have managed to create basic resources needed as a basis for end-user technologies (Scannell, 2007).

In news portals, usually news articles are pre-categorized in some common categories like latest news, economy news, world news, sport news etc. This a job that is done manually on daily basis, aiming better interface for the reader. Categorized news articles are a good source for getting classified text in case of under-resourced languages, for which there is little or no commercial interest. These categorized corpora can be used to train machine learning algorithms, that will learn how to classify a new, unknown text, according to the given trained data.

Different approaches have been used in news articles classification: text pattern mining (Chaudhari et al., 2013), which is considered to be an approach better than

the term based and phrase based approach; random forests and textual and visual multimodal features (Liparas et al., 2014); named entities, enhancing so the performing of hierarchical text classification for news articles (Guy et al., 2012); Bayesian text classification approaches like Naïve Bayes, Complementary Naïve Bayes, use of {1,2,3}-grams, and use of oversampling (Swezey et al., 2012); Dependency-LDA model which tested on large-scale datasets generally outperforms binary SVMs; three alternatives of semi-supervised learning propagation algorithms (Absorbing Random Walk, Random Walk with Restart, Local Consistency Global Consistency), aspiring to propagate political leaning of known articles and users to the target nodes (Zhou et al., 2011); Scalable Classification Algorithm for personalized news classification (Antonellis et al., 2006).

Methodology

For the purposes of this research, we have extracted the news from different news portals, taking the articles, together with their titles, links, sources, summaries, and categories. Part of the news is extracted using RSS feeds provided by the sources themselves. For those that do not offer such services, we use regular expressions to extract news and their related attributes. We extract desired text and remove unnecessary html tags. When extracting the text, there are cases when some characters are not properly encoded. All those patterns are manually detected and corrected. In this way, we have created separate (cleaned) text for categories: latest, economy, sport, showbiz, technology, culture, world. Each text is then tokenized in sentences. From the overall gained corpus, stop-words list is created. This list then is used to remove all occurrences of the stop-words, aiming better efficiency. At the end, we have created a list of tuples, where each tuple is consisted of a sentence and the category where it belongs. This list of tuples serves as input for comparison of different classification algorithms when training and testing obtained news articles. The algorithms that we have used for this comparison involve the following classifiers: Multinomial, LinearSVC, Neighbour, Bernoulli, Centroid, SGD, Perceptron, Ridge, PassiveAggressive. For this objective is used the package scikit-learn, an efficient tools for data mining and data analysis (Pedregosa et al, 2011).

To evaluate these algorithms, we use accuracy (how accurate is the classification of a model), training time (time to construct the model) and testing time (time to use the model).

Classification algorithms

Since creating a hand-labelled classified data has a high cost in terms of human resources and timing, different supervised learning algorithms are developed with goal to automatically assign a label to a new document, having initially trained the algorithm. There are a lot of such algorithms, but broadly are classified in linear classifiers, probabilistic classifiers, and vector space classifiers.

Naïve Bayes is a probabilistic classifier, where the the probability of a document *d* being in a class *c* is computed as follows:

$$P(\frac{c}{d}) \propto P(c) \prod_{1 \le k \le n_d} P(\frac{t_k}{c})$$

where n_d is length of the document, $P(t_k/c)$ is the conditional probability of term t_k occurring in a document of class c, $P(t_k/c)$ as a measure of how much evidence t_k contributes that c is the correct class, P(c) is the prior probability of c (Manning et al., 2008).

Bernoulli is an alternative of Naïve Bayes, if the data is spread according to multivariate *Bernoulli* distributions. *Multinomial* also is an alternative of Naïve Bayes, used merely in text classification, representing documents as word vector counts.

In vector space classifiers, words are axes and documents are presented as points/vectors in the vector space defined by these axes.

Centroid classifier computes a centroid for each predefined class, allocating a new document to the class which centroid is closest to this document. *Neighbour* classifier is like the centroid classifier, but it allocates a new document to the class of its nearest neighbour in the training data.

Support vector machines are another vector space based classifier that implement the following idea: input vectors are non-linearly mapped to a very high dimension feature space; in this feature space a linear decision surface is constructed (Cortes et al., 1995). *Linear Support Vector Classification* is a SVM for the case of a linear kernel.

Linear classifiers compute a linear combination or weighted sum the feature values, where the classification decision is made by comparing to a threshold:

$$\sum_i w_i x_i > \theta$$

The generalized linear model is a flexible generalization of ordinary linear regression that allows for response variables that have error distribution models other than a normal distribution. (Pedregosa et al, 2011).

Ridge regression deals with some of the complexities of least squares by forcing a penalty on the size of coefficients. These coefficients reduce a penalized residual sum of squares. *Stochastic gradient descent* classifier is a very efficient approach to fit linear models, mainly suitable once the total of samples is huge.

Perceptron is another algorithm suitable for massive scale learning. It does not require a learning rate, is not regularized (penalized), and it updates its model only on faults.

Passive/Aggressive algorithms are a type of algorithms for large-scale learning. They are like Perceptron in that they do not require a learning rate. However, opposite to Perceptron, they include a regularization parameter.

Results

We have trained and then test our articles measuring the accuracy for each classifier separately. The training/testing process is done iteratively for different number of inputs, starting from 1000 sentences, and continuing in increasing order by 1000, until the level of 20000 sentences is reached.

Table 1 illustrates the gained results for nine different approaches in classification (rows in tables). The values of a row correspond to one classification approach for twenty input sets (columns in table).

Table 1
Accuracy scores for nine classifiers, applied on twenty input sets with different size

	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}	S_{11}	S_{12}	S_{13}	S_{14}	S_{15}	S_{16}	S_{17}	S_{18}	S_{19}	S_{20}
M	0.67	0.70	0.73	0.77	0.75	0.73	0.71	0.72	0.72	0.75	0.76	0.75	0.76	0.76	0.77	0.77	0.75	0.75	0.76	0.75
L	0.74	0.79	0.82	0.85	0.86	0.87	0.85	0.87	0.87	0.90	0.90	0.89	0.90	0.90	0.91	0.91	0.91	0.91	0.91	0.92
N	0.80	0.88	0.87	0.89	0.88	0.88	0.85	0.72	0.72	0.74	0.75	0.75	0.76	0.76	0.77	0.72	0.70	0.71	0.73	0.72
B	0.53	0.65	0.67	0.70	0.69	0.72	0.73	0.75	0.76	0.81	0.78	0.78	0.80	0.80	0.82	0.82	0.82	0.82	0.83	0.83
C	0.74	0.81	0.82	0.86	0.85	0.87	0.85	0.86	0.86	0.88	0.89	0.88	0.89	0.89	0.88	0.87	0.89	0.89	0.89	0.89
S	0.77	0.84	0.86	0.88	0.88	0.89	0.87	0.87	0.87	0.89	0.88	0.87	0.87	0.87	0.89	0.88	0.87	0.87	0.87	0.87
P	0.84	0.90	0.90	0.89	0.91	0.92	0.90	0.91	0.92	0.93	0.92	0.91	0.92	0.92	0.92	0.92	0.92	0.93	0.93	0.94
R	0.73	0.77	0.80	0.84	0.85	0.85	0.83	0.85	0.86	0.88	0.88	0.87	0.88	0.88	0.90	0.90	0.89	0.89	0.90	0.91
PA	0.77	0.83	0.85	0.88	0.88	0.90	0.89	0.88	0.90	0.92	0.91	0.91	0.92	0.92	0.92	0.93	0.93	0.92	0.93	0.94
AV	0.73	0.79	0.81	0.84	0.84	0.85	0.83	0.83	0.83	0.86	0.85	0.85	0.85	0.86	0.86	0.86	0.85	0.85	0.86	0.86

*Note: The first column shortcuts: M - Multinomial, L- LinearSVC, N - Neighbour, B - Bernoulli, Centroid, S - SGD, P - Perceptron, Ridge, PA – PassiveAggressive; S_i in heading is input set with 1000*i sentences*

For each classifier, main metrics such as maximum and minimum value, average, mean and deviation are calculated.

Table 2
Main Metrics for Investigated Classifiers

Classifier	MAX	MIN	Average	Mean	Deviation
Multinomial:	0.77	0.67	0.74	0.74	0.03
LinearSVC:	0.92	0.74	0.87	0.87	0.05
Neighbour:	0.89	0.70	0.78	0.78	0.07
Bernoulli:	0.83	0.53	0.76	0.76	0.08
NearestCentroid:	0.89	0.74	0.86	0.86	0.04
SGD:	0.89	0.77	0.87	0.87	0.03
Perceptron:	0.94	0.84	0.91	0.91	0.02
Ridge:	0.91	0.73	0.86	0.86	0.05
PassiveAggressive:	0.94	0.77	0.90	0.90	0.04

The training and testing time is measured too, for each classifier and each training/testing set.

Figure 1 and Figure 2 gives the training and testing time in seconds depending from the input size.

Figure 1
Classifiers Training Time

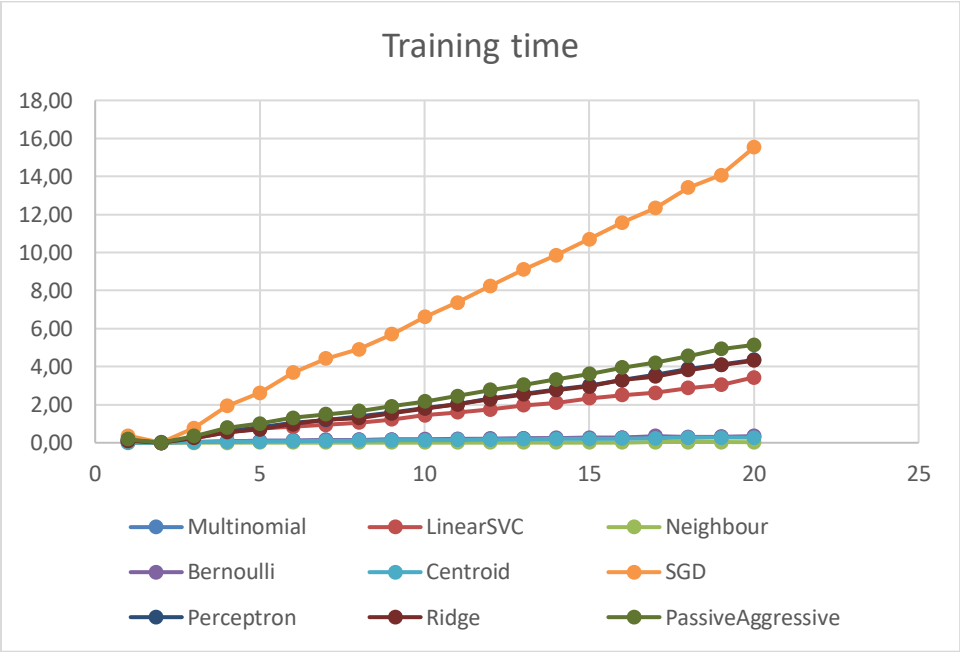
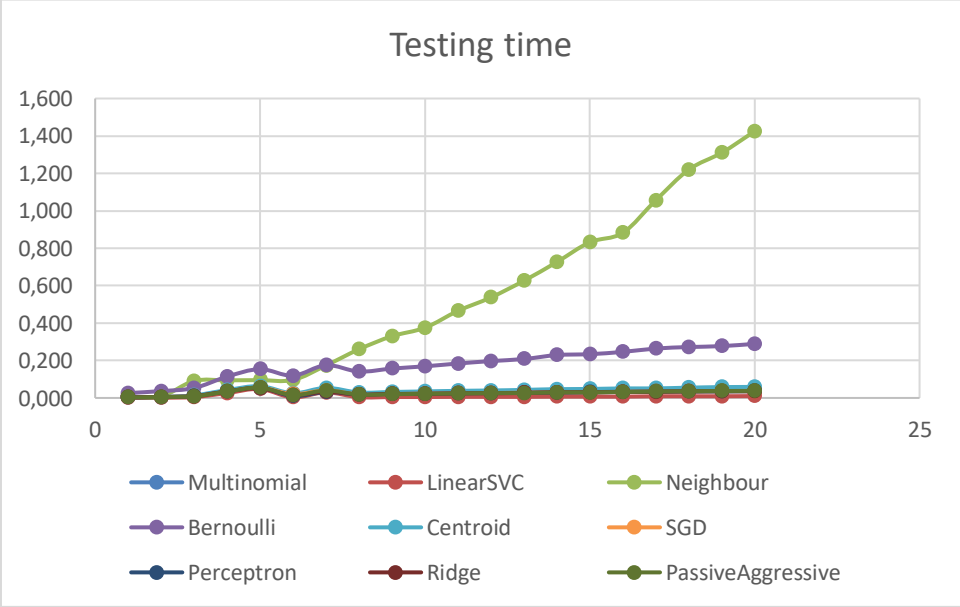


Figure 2
Classifiers Testing Time



Discussion

Table 1 shows the classifiers accuracy scores. There are some classifiers that perform nearly the same, and some perform very different in terms of accuracy. What all classifiers have in common is the fact that the accuracy increases with the input size, but even after the third input set, this increase is minimal. The last row of this table gives the average accuracy score, which for 1000 sentences is 0.73, for 2000

sentences 0.79, which is near to maximal average accuracy 0.86. It can be noticed that some of smaller sets have higher accuracy then some larger sets, but this difference is very low (0.01) and it happens just because of the random sample of the input.

Table 2 gives better understanding of the classifiers behaviour. By observing the scores in this table, it can be concluded that *Perceptron* and *Passive/Aggressive* classifiers show the highest accuracy of 0.94, but *Perceptron* has higher minimal accuracy (and the highest from the all classifiers) of 0.84, which logically results in the highest average accuracy of 0.91. This classifier has also the highest mean (0.91) and lowest deviation (0.02). From all the above, we can conclude that *Perceptron* shows the best performance in terms of accuracy in a collection of news articles, written in Albanian. The "worst" classifier in this context shows to be *Multinomial* classifier, followed by *Bernoulli* classifier.

Accuracy is the most important metric when classifying text, but when having a large input, training time is also of a vital importance. Figure 1 illustrates obtained outcome: *SGD* classifier slows rapidly when the size grows (15.52 seconds for the last set), giving an average of 7.16 seconds. This value is 4.72 seconds greater than the second slowest classifier *Passive/Aggressive* (2.44 seconds for training time). The fastest classifier for training is *Neighbour* classifier, with an average of just 0.01 seconds. Paradox ally, this classifier is the slowest when measuring the testing time (Figure 2, average 0.53 seconds). Follows *Bernoulli* classifier as the second slowest with average testing time of 0.17 seconds. LinearSVC is the fastest classifier when testing (average of 0.01 seconds). All other classifiers perform solidly with an average testing time less than 0.028 seconds. Here again, there are some small "oscillations" of the values because of the input randomness. *Perceptron* as a classifier with best accuracy results, has an average training time 2.04 seconds and testing time of 0.026 seconds.

Conclusion

Text classification is very important in different text mining applications. For under-resourced languages which have very little or no available categorized text corpora, it is difficult to investigate how different exiting classification algorithms behave in the case of a specific language. Crawling news portals is an efficient way to achieve such a benchmark. We showed that this methodology gives valuable results in classifying text written in Albanian, concluding that Perceptron gives the best performance in terms of accuracy in a collection of news articles in this language. We plan in future to extend the corpus with more pre-classified text from the web, adding new categories.

References

1. Antonellis I., Bouras C., Poulopoulos V. (2006), "Personalized News Categorization Through Scalable Text Classification", In: Zhou X., Li J., Shen H.T., Kitsuregawa M., Zhang Y. (eds) Frontiers of WWW Research and Development - APWeb 2006. APWeb 2006. Lecture Notes in Computer Science, Vol. 3841, Springer, Berlin, Heidelberg.
2. Chaudhari, S. V., Lade, Sh. (2013), "Classification of News and Research Articles Using Text Pattern Mining", IOSR Journal of Computer Engineering (IOSR-JCE), Vol. 14 No. 5, pp. 120-126.
3. Cortes, C., Vapnik, V. (1995), "Support-vector networks", Machine Learning, Vol. 20 No. 3, pp. 273–297.
4. Gui Y., Gao Z., Li R., Yang X. (2012), "Hierarchical Text Classification for News Articles Based-on Named Entities", In: Zhou S., Zhang S., Karypis G. (eds) Advanced Data

Mining and Applications. ADMA 2012. Lecture Notes in Computer Science, Vol. 7713. Springer, Berlin, Heidelberg.
5. Jurka, T.P., Collingwood, L., Boydstun, A.E., Grossman, E. and van Atteveldt, W. (2013), "RTextTools: A supervised learning package for text classification", The R journal, Vol. 5 No. 1, pp. 6-12.
6. Liparas D., HaCohen-Kerner Y., Moumtzidou A., Vrochidis S., Kompatsiaris I. (2014), "News Articles Classification Using Random Forests and Weighted Multimodal Features", In: Lamas D., Buitelaar P. (eds) Multidisciplinary Information Retrieval. IRFC 2014. Lecture Notes in Computer Science, Vol. 8849, Springer, Cham.
7. Manning, C.D., Raghavan, P., Schutze, H. (2008), Introduction to Information Retrieval, Cambridge University Press.
8. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V. Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E. (2011), "Scikit-learn: Machine Learning in Python", Journal of Machine Learning Research, Vol. 12, pp. 2825-2830.
9. Scannell, K.P. (2007), "The Crúbadán Project: Corpus building for under-resourced languages", In Building and Exploring Web Corpora: Proceedings of the 3rd Web as Corpus Workshop, Vol. 4, pp. 5-15.
10. Swezey, R. M., Sano, H., Shiramatsu, S., Ozono, T., & Shintani, T., (2012), "Automatic detection of news articles of interest to regional communities", International Journal of Computer Science and Network Security, Vol. 12 No. 6.
11. Zhou, D., Resnick, P., Mei, Q. (2011), "Classifying the Political Leaning of News Articles and Users from User Votes", International AAAI Conference on Web and Social Media, North America.

About the authors

Arbana Kadriu holds a PhD degree in Computer Sciences from Ss. Cyril and Methodius University in Skopje from 2008, with focus on natural language processing and information retrieval. She is associate professor at the Faculty of Contemporary Sciences and Technologies at SEE University in Macedonia. She has also background in artificial intelligence, machine learning, programming paradigms, software engineering and e-learning. Also, she is mentoring several master theses that involve the web information retrieval and e-learning. She is author of more than 30 research papers. Author can be contacted at **a.kadriu@seeu.edu.mk**.

Lejla Abazi-Bexheti is Associate Professor at the Faculty of Contemporary Sciences and Technologies at South East European University in Macedonia. She holds a PhD Degree in Computer Science and has been part of the CST teaching staff since 2002. During her teaching experience, she has taught courses from the area of algorithms, programming and web programming. At SEE University she was involved on resolving issue of the Learning Contents and Learning Management System. She was Director of eLearning Center at SEEU and managing Online Studies at SEEU for the period 2006-2014. Author can be contacted at **l.abazi@seeu.edu.mk**.