Data Mining using Naive Bayes classifier: an application in short news

Thais Rodrigues Neubauer
Universidade de São Paulo
Av. Arlindo Bettio, 1000
03828-000 São Paulo/SP
thais.neubauer@usp.br

Sarajane M. Peres
Universidade de São Paulo
Av. Arlindo Bettio, 1000
03828-000 São Paulo/SP
sarajane@usp.br

ABSTRACT

In the information age, a plethora of content is available on a wide range of subjects, requiring an organization capable of making that content more accessible and engaging. An interesting application of classification tasks was identified in the *Index* project, developed by the Amsterdam-based company *The Next Web*. To solve this classification task, the Naive Bayes (NB) technique was applied to classify short news in four topics. To evaluate the results produced by such a classifier, a series of tests using cross-validation were carried out. It was possible to conclude that the NB classifier had satisfactory performance, achieving about 70% of accuracy in the best cases. In this paper, we intend to present the context of the *Index* project and discuss the results obtained with the NB classifiers. Despite the good results, the project is still in progress, as it is necessary to test variations as classification techniques and text representation approaches.

CCS Concepts

•Applied computing → Document analysis; •Information systems → *Data mining;* **•Computing methodologies** → Machine learning;

Keywords

Data mining, Text classification, Statistic classification, Naive Bayes

1. INTRODUCTION

Nowadays the amount of content available grows faster than the human's ability to consume it. Every day, a large amount of information is stored to be analyzed and managed later [12]. Automatic content analysis can be even more problematic since the expectations for results from automatic analysis are rarely met. As also discussed in [13], one of the most pressing needs is the organization of information to make it more accessible and interesting to (human) readers and to automatic information retrieval mechanisms. However, precisely because of the large quantity and diversity of textual data, this task becomes challenging.

In this work, we discuss the development of a study to meet such prerogative of textual data analysis applied to the context of the *Index* project at The Next Web company. In order to investigate a feasible solution for this issue, an experiment related to short news classification was carried out using the *Naive Bayes* (NB) classifier [3], considering a multi-class classification problem (four topics). The experiments were conducted in the Natural Language Toolkit *NLTK* platform. In order to better discuss our approach, this paper is organized as follows: the theoretical background is briefly presented in Section 2; the *Index* project and the correlated textual data analysis problem are presented in Section 3; in Section 4 the approach proposed in this study is explained; the results are discussed in Section 5; and finally the conclusions and future works are outlined.

2. BACKGROUND

Currently, *text categorization* task (TC) lies mainly between the *Machine Learning* and *Information Retrieval* fields, therefore it also shares several characteristics with other areas and tasks, such as *Statistics* and *Data Mining*. The scope definition of each mentioned area is still subject of discussions. However, considering *Data Mining* as a set of tasks that, by analyzing large amounts of data and detecting the use of patterns, extracts information that may be useful, it makes sense to comprehend the TC as a *Data Mining* instance [9]. Specifically, TC consists of labeling natural language texts with pre-defined categories regarding a subject. Such a task has been studied since the 1960s, however, until the early 1990s, the topic had not been became relevant. Its relevance comes due the progressive interest in the areas and tasks related to *Data Mining* (or *Text Mining* considering textual data), as well as the availability of higher processing power [9]. The TC's relevance can be measured in terms of its application fields, which range from engineering, computer science and the biological sciences to earth sciences, social sciences and economics [12]. In order to better explain the issues related to the TC development, we present some concepts in the following sections: textual data pre-processing (Section 2.1), NB classifier (Section 2.2) and evaluation strategy (Section 2.3).

2.1 Pre-processing of textual data

In order to propose a solution in TC task, it is necessary first to pre-process the textual data and then map each document d_j to a compact (mostly numerical) representation,

SBSI 2017 June 5^{th} – 8^{th}, 2017, Lavras, Minas Gerais, Brazil

so that it can be interpreted by a classifier and by the algorithms that generate the classifier [9]. To pre-process textual data, there are some classical procedures to be applied, intending to reduce or eliminate content that is not relevant to the classification task to be performed. During this stage, it is common, and considered as good practice, to remove *stopwords*[1] and apply the process known as *stemming*[2] [9]. After the pre-process stage, the textual data is ready to be represented in a numerical form. The simplest and most common scheme of representation is the *binary representation*, in which each document d_j is represented as a vector of binary weights associated to its terms (t_k). The weight values *0* or *1* means, respectively, the absence or presence of each term t_k in each document d_j. Although there are other ways to represent a document as a vector of weights of its terms, such as *term frequency* and *term frequency - inversed term frequency* (*tf-idf*)[9], in this paper, only the binary representation is being discussed due to its simplicity.

2.2 Naive Bayes

The NB model is one of the oldest and simplest forms of the Bayesian network. It is called *naive* for assuming that, given a class, all variables or attributes are conditionally independent [8]. Thus, using the NB model, all attributes in a dataset equally influence the process of predicting a class for a given data [1]. The NB works as follows: *a priori* distribution is calculated for each class, by analyzing the training dataset; in order to classify a new datapoint (from validation or test dataset), the probability of each class is calculated considering each datapoint attribute and then all of them are combined with (weighted by) the *a priori* probabilities [11]. The result is an estimate of the probabilities of membership of a datapoint to each class. The final classification is the most probable class for that datapoint [1].

2.3 Evaluation strategy

In order to evaluate a classifier model, measuring its robustness in terms of accuracy and generalization ability, it is recommended to use strategies that are not (or are less) optimistic, capable of obtaining results close to those that would be obtained in real environments [4]. In [3], the authors present a series of strategies, and for this work, the strategy called cross-validation was chosen to be applied along with training, validation and test datasets. Although the amount of data used for this work is not in the maximum amount that can be used in the context of the *Index* project, it is possible to apply the cross-validation strategy. This strategy is able to provide a classifier quality parameter closer to that obtained in a real scenario than would be obtained using more optimistic test strategies as *holdout* [4].

3. SCOPE AND PROBLEM DEFINITIONS

The short news classifier discussed in this paper tests a new way to organize textual data of a real business project called *Index*, developed by the company The Next Web. The key idea of the *Index* is to centralize technology-related news published in a variety of digital media channels. Technology-related publications from more than one hundred different sources (*websites*) are tracked, resulting in about a thousand fresh news per day. Due to the data volume, TC arises as truly enriching, since it would associate each news with a probable topic of interest for the *Index* platform users.

The used news are stored in two different forms: *complete news*, in which a single input includes the title, excerpt and body of a news article; and *sentences*, in which the sentence is part of a news article. Some news from the *Index*'s database were manually labeled with one of the four topics of interest: *investment*, *acquisition*, *internet of things* (iot) and *virtual reality* (vr). This labeling process is the reason why TC solutions could be applied, as they are based on supervised approaches. This task generated a set with 4,145 complete news inputs: 1,214 labeled as *acquisition*, 1,193 as *investment*, 954 as *iot* and 784 as *vr*. For sentences inputs, the labeling process generated a set of 4856 inputs: 1519 labeled as *acquisition*, 1495 as *investment*, 1016 as *iot* and 826 as *vr*. Thus, with labeled short news corpora, it is possible to build a classifier capable of reaching expectations about providing a fresh, useful and complementary news organization based on its' content, meeting the interests of users.

4. PROPOSED APPROACH

To implement the NB model, we have used the functionalities provided by the framework Natural Language Toolkit (NLTK)[3] [5]. This framework allows programs written in *Python* to work with data from natural language [4]. As well as the NB model and the experiments, the pre-processing phase was also implemented with *NLTK*. The following pre-processing was performed: (i) exclusion of *stopwords*, therefore such words will not disrupt the classification process with their probable high frequency of occurrence; (ii) stemming, so that some dimensionality reduction is achieved; (iii) construction of a dictionary of *n-grams* in terms of tuples containing all *n-grams* and their respective frequencies in the whole news *corpus*; (iv) representation of each topic as the list of documents labeled with such topic, wherein each document is, in fact, a list of tuples composed by the *n-grams* [5] associated with the values 0 or 1, representing the absence or presence of the n-gram in the document (see a illustrative example in Figure 1)[6].

In order to carry out the experiments to build the classifiers, some scenarios need to be configured considering: the two different granularities of textual data (*complete news*, *sentences news*), aiming at verifying if there is difference in the classifier performance depending on the texts size; dataset balancing (*balanced*, *unbalanced*) since the complete dataset is unbalanced[7]; which *n-grams* to use, that we decided to keep the task as simple as possible using only *unigrams*), since in the literature there are no strong indications about the positive influence of the use of *bigrams* [2].

[1] *Stopwords* are words with high frequency that carry little discriminating information to help in the TC task [6].

[2] *Stemming* represent words using terms, in fact, their *radical form*.

[3] http://www.nltk.org/

[4] The NLTK offers interfaces, as *WordNet*, to facilitate handling more than 50 lexical and *corpora* resources and offers word processing libraries to divide into tokens, stem etc.

[5] N-grams: terms, separated by a hyphen, in which each term refers to stemmed text words

[6] In the experiments here presented, only unigrams (n-grams with only one term) were considered.

[7] The complete dataset has 2530 documents labeled as *acquisition*, 9330 as *investment*, 230 as *iot* and 306 as *vr*. The balanced dataset has 500 as *acquisition*, 500 as *investment*, 230 as *iot* and 306 as *vr*.

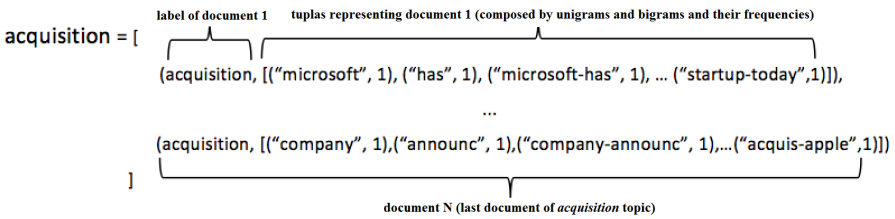

Figure 1: Example of "acquisition" topic representation (NLTK data structure)

Regarding to dataset balancing, it is important to highlight the need to nullify any influence of the randomness used to choose the datapoints to be part of the balanced dataset. Then, we carried out such random choices 30 times for each experiment scenario. All these generated balanced dataset were included in the experiments.

The next task was to define how to reduce the dimensionality of the datasets. The typical high dimensionality (thousands of dimensions) of textual datasets is due to the large number of words found in the whole set of documents being analyzed. Generally, the large number of words adds complexity to the problem and not necessarily adds discriminative and useful information to the classification models. It is known that terms that occur very frequently or very rarely in a *corpus* have a low power of description [7]. There are several ways to decrease the dimensionality of data points beyond the strategy chosen for this work, as presented in [10]. In this study, to choose which dimensions must be excluded, we analyzed the natural distribution of each term in the whole *corpus* by using boxplot graphs combined with thresholds that eliminated words with very low frequencies[8].

The information from boxplots was stored and used to perform the lower and upper cuts to reach the desired decrease of dimensionality. This was done using the minimum and maximum information of the *boxplots*, that is, the goal was the withdrawal of *outliers*. In order to minimize the amount of test (at least initially) and, at the same time, to encompass a good variety of the available search space, the values to *threshold* was empirically defined as 5, 10, and 15, based on exploratory experiments. The final amount of terms to be considered for training and testing for each dataset are shown in the Table 1.

The analysis of the *boxplots* showed that the variation of the number of terms (dimensionality), as well as the information provided in relation to the frequencies of these terms, do not differ substantially among the 30 balanced datasets generated to each variation of the other analyzed items. This fact suggests that the randomness in choosing the documents to compose balanced datasets may not influence the classifier performance, since all versions are similar regarding their terms natural distribution. Therefore, from this point on, the balancing was performed only once for each dataset, instead of 30 times as it had been done.

As we are using a binary representation for texts, the last step before training the NB models is to map the initial text representation to this one. A summary example of this representation is presented in the Figure 2.

[8]We notice that the very large number of words with low frequency represented a kind of noise that did not allow the clear interpretation of information of the boxplot graph.

Table 1: Amount of terms in each combination of data granularity, balancing and boxplot's thresholds

Data Type	Balancing	Threshold	# of terms
news	balanced	5	2248
news	balanced	10	1300
news	balanced	15	866
news	unbalanced	5	2973
news	unbalanced	10	1628
news	unbalanced	15	1113
sentences	balanced	5	789
sentences	balanced	10	386
sentences	balanced	15	175
sentences	unbalanced	5	1244
sentences	unbalanced	10	541
sentences	unbalanced	15	315

After all these decisions have been made, 12 datasets were generated by combining all of the parameters previously discussed (textual data granularity, balancing and *thesholds*). Moreover, in order to perform the *cross-validation* strategy, each dataset was divided into ten *folds*, following the recommendation in [3] and [4]. Therefore, ten different runs were performed for each of the 12 datasets. In each run, one of the *folds* was separated to perform the tests while the other nine were used for training.

5. RESULTS AND DISCUSSION

Following the procedures described in the previous section, the 12 datasets built were used to train NB classifiers and the results obtained from the performance measurement of these classifiers in the test *folds* are presented in the Table 2. The performance measurement is the average classification accuracy, calculated using the results obtained in each run of the cross-validation strategy.

From the analysis of the results presented in the Table 2, it could be noticed that the datasets with *complete news* obtained better performance than the datasets with *sentences*. Regarding to dataset balancing, there was no significant difference between the results using balanced sets and unbalanced sets. For threshold values, there was no significant difference between the results of the combination involving *complete news*, but in the combinations with *sentences*, the results obtained for the lower threshold values were better than for the higher threshold values.

In order to check if the cross-validation strategy is robust for the discussed problem, some tests were made by

```
acquisition = [
        (acquisition, [("microsoft", True), ("has", True), ... ("wait-home", False)]),
                                        ...
        (acquisition, [[("microsoft", False), ... , (company-announc", 1), ... , ("wait-home", False)])
]
```

Figure 2: Example of "acquisition" topic with binary representation (NLTK data structure)

Table 2: Results for each dataset and considering each topic

Data type	Balancing	Threshold	Accuracy acquisition	Accuracy investment	Accuracy iot	Accuracy vr
news	balanced	5	**0.6400**	**0.7500**	**0.6600**	**0.7950**
news	balanced	10	0.6050	0.7250	**0.6700**	**0.8150**
news	balanced	15	0.6000	0.7200	0.6550	**0.8100**
news	unbalanced	5	**0.6580**	**0.7600**	0.6435	0.7111
news	unbalanced	10	0.6180	**0.7580**	0.6435	0.7344
news	unbalanced	15	0.6000	**0.7540**	**0.6739**	0.7183
sentences	balanced	5	0.6038	0.6963	0.5662	0.6725
sentences	balanced	10	0.4825	0.6850	0.5062	0.5900
sentences	balanced	15	0.4188	0.7087	0.4225	0.4975
sentences	unbalanced	5	0.6618	0.7543	0.5794	0.6318
sentences	unbalanced	10	0.5802	0.7376	0.5156	0.5760
sentences	unbalanced	15	0.5733	0.7522	0.4461	0.5121

generating 10 different classifiers to one of the datasets using, for each classifier, 80% of its datapoints (randomly chosen) to train the models and the other 20% to test it. By analysing the results, we noticed that the standard deviation among the accuracies obtained in each classifier using such a strategy was significantly greater than the deviation observed among runs of the cross-validation strategy. Therefore, it was possible to affirm that the cross-validation in fact brought greater reliability to the results.

6. CONCLUSION AND FUTURE WORK

The NB classifier discussed in this paper was trained with real data of the project *Index* from The Next Web company. Considering the data context and the initial expectations of the company, the results achieved (about 70% of accuracy) were satisfactory. Despite the good results, it is needed and planned to test the classifier in the real production environment. The next steps planned to this work include: tests with other classification techniques, as neuro-fuzzy systems, which are able to take advantage of the robustness of neural networks and the modeling flexibility of fuzzy set theory; exploration of other textual data representation. The goal with further studies is to improve accuracy rates.

7. ACKNOWLEDGEMENTS

We are grateful to all The Next Web members, especially Patrick de Laive and Otto Rottier, for data access and for helping in first steps of building the classifier.

8. REFERENCES

[1] S. Bird, E. Klein, and E. Loper. *Natural Language Processing with Python*. O'Reilly Media, 2009.

[2] I. A. Braga, M. C. Monard, and E. T. Matsubara. Combining unigrams and bigrams in semi-supervised text classification. pages 489–500, 2009.

[3] J. Han, M. Kamber, and J. Pei. *Data Mining: Concepts and Techniques*. Morgan Kaufmann, 3rd edition, 2011.

[4] R. Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 2*, IJCAI'95, pages 1137–1143, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc.

[5] E. Loper and S. Bird. Nltk: The natural language toolkit. In *Proc. of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Lang. Proc. and Comp. Ling. - V.1*, pages 63–70. Assoc. for Comp. Ling., 2002.

[6] M. Makrehchi and M. S. Kamel. *Automatic Extraction of Domain-Specific Stopwords from Labeled Documents*, pages 222–233. Springerg, 2008.

[7] C. J. V. Rijsbergen. *Information Retrieval*. London: Butterworths, 2 edition, 1979.

[8] S. J. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Pearson, 2 edition, 2003.

[9] F. Sebastiani. Machine learning in automated text categorization. *ACM Comput. Surv.*, 34(1), Mar. 2002.

[10] W. Shalaby, W. Zadrozny, and S. Gallagher. Knowledge based dimensionality reduction for technical text mining. In *2014 IEEE International Conference on Big Data*, pages 39–44, 2014.

[11] L. A. Silva, S. M. Peres, and C. Boscarioli. *Introdução à Mineração de Dados com Aplicações em R*. Elsevier, 1st edition, 2016.

[12] R. Xu and D. Wunsch, II. Survey of clustering algorithms. *Trans. Neur. Netw.*, 16(3):645–678, 2005.

[13] X. Zhang and B. Wu. Short text classification based on feature extension using the n-gram model. In *2015 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*, pages 710–716, 2015.