# Text Classification in the Field of Search Engines

Javier Pérez[1], David Martínez[1],Darío Arenas[1], David Llorente[1], Eugenio Fernández[1], Antonio Moratilla[1],
[1](Computer Science/ University of Alcalá, Spain)

***ABSTRACT :*** *Search and communication are the most popular uses of the computer. Not surprisingly, many people in companies and universities are trying to improve search by coming up with easier and faster ways to find the right information.Behind this whole business model underlies a problem of text classification. Classify the intention that users reflect through query to provide relevant results in terms of both organic search results and sponsored links.Inspired by stock market machine learning systems, a text classification tool is proposed. It consists of using a combination of classic text classification techniques to select the one that offers the best results according to an established machine learning criterion.*

***KEYWORDS –****Text Classification, Naive Bayesian Classifier, Random Forest, Support Vector Machine, Cross-validation, Schwarz Criterion.*

## I. INTRODUCTION

With the rise of the Internet the large amount of information available made searching for information a very complex task. The enormous limitations implied by a human being crawling around all the information on the Internet meant that a site could not be found for its real keywords, but had to be interpreted and searched based on the descriptors that each directory used to classify the different pages ([1],[2]).

The need for a search system led to the development of search engines. The first was developed in 1993 at MIT by Mathew Gray: Wandex. It was a Perl-based crawling robot capable of reading urls and indexing them to find them quickly. Later others would arrive, such as Aliweb or Altavista. Aliweb was launched in 1994 and showed a remarkable improvement by indexing webs based on the words that appeared in his domain. Aliweb did not use a web robot, but instead depended on being notified by website administrators of the existence at each site of an index file in a particular format ([1],[2]).

Thereafter other search engines such as Altaweb would arrive, using variations of the same concept. Deploying robots to retrieval index not only text, but also multimedia documents with structure, significant text content, and other media images, videos, news or scientific articles [1].

Many other search engines were launched but after the dot-com bubble only a handful of them survived. In 2000 Google's search engine started to stand out. The premise of its PageRank algorithm was to establish a ranking based on the idea that the best websites would be more linked than others.By the same date Yahoo acquired the company Inktomi and later Overture. Combining the technologies of both, he created his own search engine and launched it on the market in 2004 [2].

Using the Inktomi algorithm, Microsoft launched MSN Search in 1998. Finally, and after developing its own web crawler, Microsoft renamed its search engine as Bing in 2009. That same year, Yahoo agreed with Microsoft to use Bing technology to powered Yahoo Search! leaving Google and Bing as the main search engines in the market[3].

In order to finance their services, search engines developed their own business models, especially pay-per-click programs. They started being offered by Open Text in 1996, although they became popular with the rest of search engines. In 2000 Google began to offer ads on its search results pages through the AdWords program [3].

Pay-per-click programs were consolidated as the main form of search engine monetization in 2007. In a market dominated by Google, Yahoo and Microsoft formed the Search Alliance in 2010.

Yahoo and Bing offer equivalent services called Yahoo Gemini and Bing Ads[2].

The search engine business is based on the sale of advertising. Classifying the intention that the user expresses by means of query is fundamental to offer sponsored links that are really interesting. On the other hand, classifying ads correctly automatically is also a very interesting task [4].

## II. TEXT CLASSIFICATION

Text analytics refers to the process of automatically extracting high-value information from text. This extraction usually involves a process of structuring the input text, discovering patterns in the structured text and finally evaluating and interpreting the results.

Classification or categorization is a branch of text classification that consists of assigning a text to one or more categories from a predefined taxonomy, taking into account the overall content of the text.

Numerous tools are available for the classification of texts. For this study we will focus on the following three: Naïve Bayes, Random Forest and Support Vector Machines.

### 2.1 NAÏVE BAYES

The Naïve Bayesian classifier is a probabilistic classifier based on Bayes' theorem and several simplifying hypotheses. The term naive comes from the assumption that the presence or absence of a particular characteristic is not related to the presence or absence of any other characteristic ([5],[6])

Bayes Naïve classifiers have the advantage of being able to be trained very efficiently in supervised learning environments. In addition, it can be used for many practical applications without accepting Bayesian probability ([5],[7]).

The implementations of the naïve Bayesian classifiers only require a small amount of training data to estimate the parameters (means and variances of variables) needed for classification. As the independent variables are assumed, it is only necessary to determine the variances of the variables of each class and not the entire covariance matrix [8].

All model parameters (e. g., prioris classes and characteristics of probability distributions) can be approximated with relative frequencies of the training set([6],[9]). These are the maximum likelihood estimates of probabilities.

A prior class can be calculated by assuming comparable classes or by calculating an estimate of the class probability of the training set [10]. In order to estimate the parameters of the distribution of a characteristic, it is necessary to assume a distribution or generate nonparametric statistical models of the characteristics of the training set [11].

### 2.2 RANDOM FOREST

Random forest is a continuous learning technique that can be used for classification among other tasks. It is a combination of classification trees such that each tree depends on the values of an independently tested random vector with the same distribution for each of these [12]. It is a substantial modification of bagging that builds a long collection of uncorrelated trees and then averages them [13].

A classification tree can be considered as the structure resulting from the recursive partitioning of the representation space from the training set. This recursive partition results in a hierarchical organization of the representation space [14]. If we consider each space question/partition as a parent node and each output space as a child node, we get the tree structure.

Pattern classification is based on a series of questions about the values of the text attributes. Starting from the root node and following the path determined by the answers to the questions of the internal nodes, until reaching a leaf node. The label assigned to this leaf is the one that will be assigned to the pattern to be classified ([13,15]).

The basic idea of bagging is to average many noisy but approximately impartial models to reduce variation. Trees are the ideal candidates for bagging because they can record complex interaction structures in data with low bias if they are deep enough. Since the trees are particularly noisy, they benefit greatly from averaging [16].

Among the advantages of this algorithm is that its performance in some applications is equivalent to boosting but much easier to train and adjust [17].

## 2.3 SUPPORT VECTOR MACHINES

Support Vector Machines (SVMs) are a set of supervised learning algorithms related to classification and regression problems. Given a set of sample training examples we can label the classes and train an SVM to build a model that predicts the class of a new sample ([18],[19]).

Intuitively, a SVM is a model that represents the sample points in space, separating the classes in spaces by using a separation hyperplane defined as the vector between the two points, from the two closest classes (support vector) [20]. Subsequent samples will be placed in correspondence with this model. Their belonging to one class or another will be determined by the space to which they belong.

More formally. Given a set of points, subset of a larger set (space), in which each of them belongs to one of two possible categories, an algorithm based on SVM builds a model capable of predicting whether a new point belongs to one category or the other. As in most supervised classifying methods, the input data (points) are seen as p-dimensional vectors [21].

The SVM is looking for a hyperplane that optimally separates the points of one class from that of another, which may have been previously projected to a superior dimensional space ([21],[22]).

Ideally, the SVM-based model should produce a hyperplane that completely separates the data from the studied universe into two categories. However, this is not always possible or desirable, since it may be a non-generalizable model for other data (overfitting) [20].

In order to add some flexibility, SVMs employ a C parameter that controls compensation between training errors and rigid margins. In this manner, a soft margin that allows some classification errors and at the same time penalises them is defined [23].

## III. ADDITIONAL TECHNOLOGIES

Our model requires the use of supplementary technologies. On the one hand, a model that improves the quality of training: cross-validation. In addition, we will have to add a technology that determines the best result among the three classification algorithms: the Schwarz Criterion.

## 3.1 CROSS-VALIDATION

Cross-validation is a technique used to evaluate the results of statistical analysis and ensure that they are independent of the partition between training and test data. It consists of repeating and calculating the arithmetic mean obtained from the evaluation measures on different partitions [24]. It is used in environments where the main objective is prediction and we want to estimate the accuracy of a model that will be carried out in practice.

Cross-validation comes from the improvement of the holdout method. This consists of dividing the sample data into two complementary sets, performing the analysis of one subset (training data) and validating the analysis with the other subset (test data) [25]. In this way, the approximation function is only set with the training dataset and calculates the output values for the test dataset.

In K-fold cross-validation, the sample data is divided into K subsets. One of the subsets is used as test data and the rest (K-1) as training data. The cross-validation process is repeated during k iterations, with each possible subset of test data. Finally, the arithmetic mean of the results of each iteration is performed to obtain a single result [26].

Although this method is expensive from a computational point of view, the result is very accurate as it is evaluated from k combinations of training and test data [24].

## 3.2 SCHWARZ CRITERION

The Bayesian Information Criterion (BIC) or Schwarz Criterion (SBIC) is a measure of accuracy of fit of a statistical model. It is often used as a criterion for model selection among a finite set of models [27].

It is based on the logarithmic probability function (LLF) and is closely related to the Akaike Information Criterion (AIC). As in AIC, SBIC introduces a penalty term for the number of parameters in the model, but the penalty is higher than when using AIC ([27],[28]).

In general, the SBIC is defined as [29]:

$$SBIC = k \times \ln n - 2 \times \ln L$$

Where:

- k: is the number of parameters of the model.
- $\ln L$ : is the log-verosimilitude function

Given two estimated models, the model with the lowest SBIC value is preferred; a low SBIC implies a smaller number of explaining variables, better fit, or both [30].

## IV. PROPOSAL OF SOLUTION

The development of three text classification tools using the following methods: Naive Bayes, Random Forest and Vector Support Machines has been proposed.

Each of the above-mentioned techniques will be trained and tested. It will use a set of 2.7 billion keywords classified into 22 categories that will be divided into training and testing. We will use a cross-validation technique to ensure that both sets are statistically consistent.

Once we have a model trained and validated for the production of each of the three indicated technique we will need to choose the best option.

The selection of the "optimal" model from multiple candidate models represents a major challenge to properly interpret the data. This selection can be made by using the Akaike and Schwarz criteria for information, the verisimilitude quotient test and the criteria based on the estimated responses ([31],[32]).

We are going to create a model that makes use of Schwarz Criterion to choose the answer that offers the best results.

## V. CONCLUSION

The classification of text represents a classic problem within the realm of artificial intelligence. With the explosion of the Internet, the need for optimal classification in terms of time and quality is even greater.

In order to classify the intentions expressed by a user of a search engine to identify the topics that are of interest to him/her, a model has been proposed that combines several techniques.

A model using Schwarz's criterion has been proposed to choose the best response from those offered by three different algorithms: naive Bayesian classifier, random forest and a vector support machine. For training and testing of this model, we have had a dataset of 2.7 billion queries on which the cross-validation techniques have been applied.

## REFERENCES

[1] *Seymour, T.; Frantsvog, D.; Kumar, S. (2011). International Journal of Management and Information Systems; Littleton Vol. 15, Iss. 4: 47-58.*

[2] *Schwartz, C. (1998). Web search engines. Journal of the American Society for Information and Science. Volume 49, Issue 11, Pages 973–982.*

[3] *Liawa, S.S.; Huangb, H.M. (2003). An investigation of user attitudes toward search engines as an information retrieval tool. Computers in human behavior. Volume 19, Issue 6, Pages 751-765.*

[4] *Croft, W.B.; Metzler, D.; Strohman, T. (2010). Search engines: Information retrieval in practice. Pearson. ISBN-13: 978-0136072249. pp 233-250.*

[5] *Domingos, P.; Pazzani, M. (1997). On the optimality of the simple Bayesian classifier under zero-one loss. Machine Learning, 29: 103–137.*

[6] *Webb, G. I.; Boughton, J.; Wang, Z. (2005). Not So Naive Bayes: Aggregating One-Dependence Estimators. Machine Learning. Springer. 58 (1): 5–24.*

[7] *Mozina, M.; Demsar, J.; Kattan, M.; Zupan, B. (2004). Nomograms for Visualization of Naive Bayesian Classifier. Proc. PKDD-2004. pp. 337–348.*

[8] *Maron, M. E. (1961). Automatic Indexing: An Experimental Inquiry. Journal of the ACM. 8 (3): 404–417.*

[9] *Minsky, M. (1961). Steps toward Artificial Intelligence. Proc. IRE. 49. pp. 8–30.*

[10] *Lewis, D.; Naive (Bayes) at Forty: The Independence Assumption in Information Retrieval, Proceedings of the 10th European Conference on Machine Learning, p.4-15, April 21-23, 1998.*

**Page 4**

[11] *Keogh, E.; Pazzanni, M. (1999). A comparison of distribution-based and classification-based approaches. Proceedings Artificial Intelligence & Statistics 1999.*

[12] *Breiman, L. (2001). Random Forests. Machine Learning 45 (1): 5–32.*

[13] *Ho, T. (1998). The Random Subspace Method for Constructing Decision Forests. IEEE Transactions on Pattern Analysis and Machine Intelligence 20 (8): 832–844.*

[14] *Lin, Y.; Jeon, Y. (2006). Random forests and adaptive nearest neighbors. Journal of the American Statistical Association. 101 (474): 578–590.*

[15] *Geurts, P.; Ernst, D.; Wehenkel, L. (2006). Extremely randomized trees. Machine Learning. 63: 3–42.*

[16] *Amit, Y.; Geman, D. (1997). Shape quantization and recognition with randomized trees. Neural Computation 9 (7): 1545–1588.*

[17] *Ham, J.; Chen, Y.C.; Crawford, M.M.; Ghosh, J.; (2005). Investigation of the random forest framework for classification of hyperspectral data. EEE Transactions on Geoscience and Remote Sensing 43, 492–501*

[18] *Breiman, L. (2001). Random Forests. Machine Learning 45 (1): 5–32.*

[19] *Ho, T. (1998). The Random Subspace Method for Constructing Decision Forests. IEEE Transactions on Pattern Analysis and Machine Intelligence 20 (8): 832–844.*

[20] *Lin, Y.; Jeon, Y. (2006). Random forests and adaptive nearest neighbors. Journal of the American Statistical Association. 101 (474): 578–590.*

[21] *Geurts, P.; Ernst, D.; Wehenkel, L. (2006). Extremely randomized trees. Machine Learning. 63: 3–42.*

[22] *Amit, Y.; Geman, D. (1997). Shape quantization and recognition with randomized trees. Neural Computation 9 (7): 1545–1588.*

[23] *Ham, J.; Chen, Y.C.; Crawford, M.M.; Ghosh, J.; (2005). Investigation of the random forest framework for classification of hyperspectral data. EEE Transactions on Geoscience and Remote Sensing 43, 492–501.*

[24] *Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence. San Mateo, CA: Morgan Kaufmann. 2 (12): 1137–1143.*

[25] *Efron, B.; Tibshirani, R. (1997). Improvements on cross-validation: The .632 + Bootstrap Method. Journal of the American Statistical Association. 92 (438): 548–560.*

[26] *Trippa, L.; Waldron, L.; Huttenhower, C.; Parmigiani, G. (2015). Bayesian nonparametric cross-study validation of prediction methods. The Annals of Applied Statistics. 9 (1): 402–428.*

[27] *Schwarz, G. E. (1978). Estimating the dimension of a model. Annals of Statistics, 6 (2): 461–464.*

[28] *Wit, E.; Heuvel, E.; Romeyn, J.W. (2012). 'All models are wrong...': an introduction to model uncertainty. StatisticaNeerlandica. 66 (3): 217–236.*

[29] *Kass, R.E.; Raftery, A. E. (1995). Bayes Factors. Journal of the American Statistical Association, 90 (430): 773–795.*

[30] *Kass, R.E.; Wasserman, L. (1995). A Reference Bayesian Test for Nested Hypotheses and its Relationship to the Schwarz Criterion, 90 (431): 928–934.*

[31] *Wang, J.; Schaalje, G.B. (2009). Model selection for linear mixed models using predictive criteria. Communications in Statistics. Simulation and Computation, 38, 788–801.*

[32] *Allen, D. M. (1974). The relationship between variable selection and data augmentation and a method of prediction. Technometrics, 16, 125–127.*