Article **Journal of Social Researches**
June 2022, Vol.8 No.21 27-33

Automatic document classification using machine learning

Clasificación automática de documentos usando aprendizaje automático

GUZMÁN-CABRERA, Rafael†*

ID 1st Author: *Rafael, Guzmán-Cabrera* / ORC ID: 0000-0002-9320-7021, **Researcher ID Thompson:** L-1158-2013, **Scopus ID Author:** 56002744300, **CVU CONACYT ID:** 88306

DOI: 10.35429/JSR.2022.21.8.27.33 Received January 30, 2022; Accepted June 30, 2022

Abstract

In many areas of professional development, the categorization of textual objects into predefined categories is used. In this paper we present a description of the automatic classification of documents, as well as the way in which this task is evaluated. The results of experiments carried out with a set of plain text files, corresponding to news items referring to five categories of natural disasters in Spanish, are shown. Two classifiers were built, one based on support vector machine and the classical Bayesian classifier. Different percentages of the file set were used to build the classifiers (10, 30 and 70%) and the rest was used to test the classifier. The best results are obtained for the SVM-based classifier with 99.24% of correctly classified instances.

Text classification, SVM, Bayes, Evaluation

Resumen

En muchas áreas de desarrollo profesional es empleada la categorización de objetos textuales en categorías previamente definidas. En este trabajo se presenta una descripción de la clasificación automática de documentos, así como la manera en cómo se evalúa esta terea. Se muestran resultados de experimentos realizados con un conjunto de archivos en texto plano, correspondientes a noticias referentes a cinco categorías de desastres naturales en español. Se construyeron dos clasificadores, uno basado en maquina de vectores de soporte y el clásico clasificador bayesiano. Se utilizaron diferentes porcentajes del conjunto de archivos para construir los clasificadores (10, 30 y 70%) y el resto se utilizo para la prueba de este. Los mejores resultados se obtienen para el clasificador basado en SVM con un 99.24% de instancias clasificadas correctamente.

Clasificación de textos, SVM, Bayes, Evaluación

Citation: GUZMÁN-CABRERA, Rafael. Automatic document classification using machine learning. Journal of Social Researches. 2022. 8-21:27-33.

* Correspondence to Author (E-mail: guzmanc@ugto.mx)
† Researcher contributing as first author.

Introduction

Classification or categorisation is the task of assigning a set of objects to two or more predefined classes or categories. In many areas of professional development, categorisation of new objects is employed. This process is costly and time consuming [1]. The classification problem can be divided into two parts: training and classification. Learning involves the acquisition of general concepts from a set of training examples.

One approach to building a text categorisation system is to manually assign a set of documents to be categorised. In this case the hierarchies or subject areas are assigned by an expert. However, this process is usually very costly and time-consuming, since an expert is needed for each area or application in which the classification task is to be carried out, and a change of area implies the need for a new expert to define the categories or the documents that belong to each category as well as the rules that allow decisions to be made about new documents to be classified [2].

Although it is possible to build a text classification system manually, the most widely used approach today is to use information retrieval and machine learning techniques to induce a classification model, as in [3-5]. Learning-based systems are also faster to build than rule-based or language model-based systems.

Much of the research developed has been applied to binary problems, where a document is classified as relevant or not relevant with respect to predefined topics. However, there are many text data sources such as news, e-mail and digital libraries, just to mention a few, which are composed of different topics and represent a multi-class categorisation problem. In addition to multi-class classification there are other factors that increase the complexity of classification, such as some natural language features like synonymy, ambiguity and skewed distributions, which make the classification task more difficult [6].

Document classification can be seen as the task of assigning a value of 0 or 1 to each element of a decision matrix. Where the documents to be classified are represented by the set D = {d1,..., dm}, while the set of possible categories to assign to the set of documents is represented by C = {C1,..., Cm}. In this way, a value a i j = 1 would be interpreted as the document di belonging to the category c j. Figure 1 shows the scheme used for the categorisation of documents.

	Documents to classify							
predefined categories		d_1			d_j			d_n
	C_1	a_{11}	…	…	…	…	…	a_{1n}
	…	…	…	…	…	…	…	…
	C_i	a_{i1}			a_{ij}			a_{in}
	…	…	…	…	…	…	…	…
	C_m	a_{m1}			a_{mj}			a_{mn}

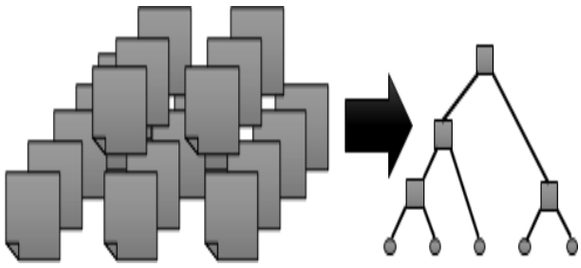

Figure 1 Categorisation of documents

In the context of machine learning, classification is one of the following two steps:

- From a set of observations, classifying consists of establishing the existence of classes or groups of data (unsupervised learning).
- Knowing the existence of certain classes, classifying consists of establishing a rule to place new observations in one of the existing classes (supervised learning).

The classification task can be carried out in two ways, the first consists of assigning exactly one category to each document, while the second consists of assigning each category to a document (each element of C is assigned an element of D). By assigning one row at a time of the matrix we have CPC (Category-Pivioted-Categorisation). It is more common to assign rows (CPC) than columns (DPC) in the categorisation task.

ISSN: 2414-4835

The rest of the article is organised as follows: Section 2 presents two of the existing classification methods using machine learning techniques: the Bayesian classifier and support vector machines (SVM). Section 3 presents the evaluation measures of text, accuracy, recall and fallout classification systems. Section 4 presents a description of the experiments performed, as well as the results obtained, and finally section 5 presents the conclusions and future work.

1 Machine learning based classification techniques

The Bayesian classifier (Bayes, 1764) is considered as part of the probabilistic classifiers, which are based on the assumption that quantities of interest are governed by probability distributions, and that the optimal decision can be made by reasoning about these probabilities together with the observed data. In tasks such as text classification, this algorithm is among the most commonly used. The naive Bayes algorithm uses the training set to estimate the parameters of a probability distribution describing the training set. The document with the highest probability is assigned the category. In this scheme the classifier is constructed by estimating the probability of each class, which is represented by Tr. Then, when a new instance ij is presented, the classifier assigns the most likely category c ϵ C, after applying the rule $c = \arg max_{c_i \epsilon C} p(c_i|i_j)$, and using Bayes' theorem to estimate the probability we have:

$$c = \arg max_{c_i \epsilon C} \frac{p(i_j|c_i)p(c_i)}{p(i_j)}$$

Considering that the denominator of this equation does not change between categories, we have:

$$c = \arg max_{c_i \epsilon C} p(i_j|c_i)p(c_i)$$

Taking into account that the scheme is called "naive" due to the assumption of independence between attributes, i.e., it is assumed that the features are conditionally independent given the classes.

This simplifies the calculations by producing:

$$c = \arg max_{c_i \epsilon C} p(c_i)\Pi_{k=1}^{n} p(a_{kj}|c_i)$$

Where p(c $_i$) is the fraction of examples in Tr belonging to class c i, and $p(a_{kj}|c_i)$ is calculated according to Bayes' theorem. In summary, the learning task in the naive Bayes classifier consists of constructing a hypothesis by estimating the different probabilities $p(c_i)$ y $p(a_{kj}|c_i)$ in terms of their frequencies over T_r.

In tasks such as text classification, this algorithm is among the most widely used [7-8]. A basic guide to the different directions that naive Bayes research has taken, which are characterised by modifications made to the algorithm, is presented in [7].

SVM support vector machines have been shown to achieve good generalisation performance on a wide variety of classification problems, most recently on problems such as text classification [9] and [10], where SVM tends to minimise generalisation error (classifier error on new instances). In geometric terms, SVM can be seen as the attempt to find a surface (σ_i) that separates positive examples from negative ones by the widest possible margin. The search for σ_i that satisfies that the minimum distance between it and a training example is maximal is performed through all surfaces σ_1,σ_(2,...) in the A-dimensional space that separate the positive examples from the negative ones in the training set (known as decision surface). The best decision surface is determined only by a small set of training examples, called support vectors.

An important advantage of SVM is that it allows the construction of non-linear classifiers, i.e., the algorithm represents non-linear training data in a high-dimensional space (called "feature space"), and constructs the hyperplane that has the maximum margin. Furthermore, it is possible to compute the hyperplane without explicitly representing the feature space. In tasks such as text classification, this algorithm is among the most widely used [7-8]. A basic guide to the different directions naive Bayes research has taken, which are characterised by modifications made to the algorithm, is presented in [7].

ISSN: 2414-4835

SVM support vector machines have been shown to achieve good generalisation performance on a wide variety of classification problems, most recently on problems such as text classification [9] and [10], where SVM tends to minimise generalisation error (classifier error on new instances). In geometric terms, SVM can be seen as the attempt to find a surface (σ_i) that separates positive examples from negative ones by the widest possible margin. The search for σ_i that satisfies that the minimum distance between it and a training example is maximal is performed across all surfaces σ_1,σ_(2,...) in the A-dimensional space that separate the positive examples from the negative ones in the training set (known as decision surface). The best decision surface is determined only by a small set of training examples, called support vectors.

An important advantage of SVM is that it allows the construction of non-linear classifiers, i.e., the algorithm represents non-linear training data in a high-dimensional space (called "feature space") and constructs the hyperplane that has the maximum margin. Furthermore, it is possible to compute the hyperplane without explicitly representing the feature space.

2 Evaluation

Within the performance of a learning system, one of the most important factors is the measurement of the acquired knowledge that will enable the system to perform the classification task. If the learning system has access to the input and output, it is referred to as supervised learning, if it only has access to the output, it is referred to as reinforcement learning, while if it has no access to any information about the output, it is referred to as unsupervised learning. The following are the most commonly used measures of the performance of classification systems.

A system is said to learn from its experience E, with respect to some kind of task T and a performance measure P if the performance of the program in performing the tasks T, improves with experience E, according to the measure P.

To improve the characteristics of a learning system, the following factors must be taken into account:

- Exact type of knowledge to be learned.
- Knowledge representation (usually a set of weighted rules that will allow to make the assignment of the most probable category).
- Learning mechanism.

For a binary classification, typically classifiers are evaluated using a contingency table as shown in table 1 [11].

		Human classification		
System decision		Yes	No	
	Yes	a	b	$a+b$
	No	c	d	$c+d$
		$a+c$	$b+d$	$a+b+c+d$

Table 1 Contingency table for evaluation of classification systems

Each entry in the table specifies the number of decisions of a particular type. For example "b" is the number of false positives, i.e. the system classifies it as "yes", but the human expert classifies it as "no". Among the most important measures that allow us to measure the performance of classification systems we have:

Precision and Recall these measures are also used in information retrieval tasks, where they represent the proportion of retrieved documents that are relevant to a given request or query. They are defined as:

$$accuracy = \frac{a}{a+b}$$

$$Recall = \frac{a}{a+c}$$

Accuracy represents the confidence level of the classifier, usually represented as the proportion of correct classifications it is able to produce. Accuracy is measured with respect to data other than the training dataset.

The proportion of non-relevant documents that are retrieved can be obtained by means of the evaluation measure called Fallout, which is defined by:

$$Fallout = \frac{b}{b+d}$$

ISSN: 2414-4835

Another evaluation measure used is the classification accuracy, which allows us to know the proportion of objects classified correctly and is given by:

$$accuracy = \frac{a+b}{a+b+c+d}$$

However, the contingency table has some limitations as, for example, it does not take into account the possibility that different errors have different costs, which requires more general decision theory modelling. In addition, it requires all inputs to be binary. However, it would be desirable to assign a weight to each category in the table and then discuss an evaluation approach for this case.

Another way to measure the effectiveness of a ranking system is by means of micro and macro averages. For a set of q queries and d documents a total of n = q*d decisions are taken. Micro averaging considers the q*d decisions as a single group and calculates precision, recall and fallout as defined above. Whereas macro averaging does this separately for the set d of documents associated with each query and subsequently calculates the measure of q results obtained. The difference between these two measures is that macro averaging gives equal weight to each category while micro averaging gives equal weight to each object. The two types of averages can give different results when the precision is averaged over categories of different sizes. Accuracy determined by micro averaging is called out for large categories, while accuracy determined by macro averaging provides a better sense of quality or classification across all categories.

3 Description of experiments and results

All documents must be transformed into an internal expression for text search methods to be able to use them. One of the most common representations is the vector representation where the dimension of the vector corresponds to the terms occurring within the training and the value of each individual entry corresponds to the weight of the term in question in the document. Normally the weight of these words is reflected in the semantic importance that these words have in the document in which they occur and are automatically computed by weighing functions.

The aim is to have words that are highly discriminative as classification attributes, i.e. words that allow to separate one class from another. In this sense, words that occur only in documents belonging to one class will be more relevant than words that occur in documents belonging to different classes.

These techniques tend to generate very large vectors, often with more than a thousand elements. Because of this it is common to find techniques to reduce the dimensionality of the vectors before starting the construction of the internal representation of the documents, that is, a new vector is generated with a new space in which the representation of the document is such that the new vector has a much smaller number of dimension than the original vector, an important class of techniques are feature extraction methods. Feature extraction methods define a new vector space in which each dimension is a combination of some or all of the original dimensions.

Many of these dimensionality reduction functions are based on statistical measures, e.g., chi-square, mutual information and information gain among others.

The files provided to carry out the classification task were 439 from 5 different categories, which were downloaded from the Fuerza informativa azteca website and the newspaper reforma, table 2 shows the number of files downloaded per category.

Files to classify	
Forestry	92
Inundation	87
Earthquake	143
Hurricane	76
Drought	41

Table 2 Number of files in the corpus to build and test the classifier

Figure 2 shows the diagram governing the experiments performed, on the left side is the learning part and on the right side enclosed in dotted lines is the testing part. The first thing we do is to learn the characteristics of each category so that the system is able to assign a category to a new document.

ISSN: 2414-4835

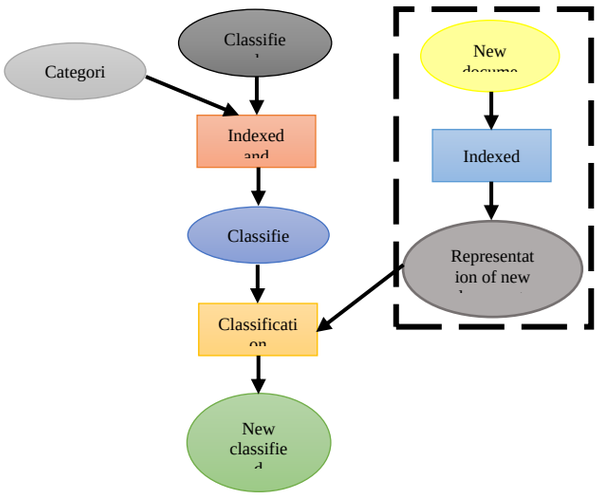

Figure 2 General diagram of the classification system implemented

As can be seen in Figure 2, the new documents to be classified go through a pre-processing stage (indexing and representation of the new documents). This stage aims to reduce the size of the documents by eliminating the parts that are not relevant for predicting the content. This is achieved by removing: HTML tags and punctuation symbols. In addition, stopwords are removed and stemming is carried out. Figure 3 illustrates this process.

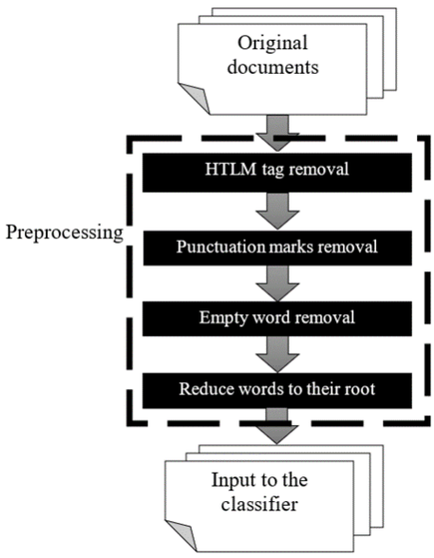

Figure 3 Pre-processing applied to new documents

Once the documents have been pre-processed, feature extraction and indexing of the documents is performed. Indexing is the representation of the documents as a feature vector. At this point all the vocabulary of the existing examples in the learning corpus was collected, resulting in a total of 60503 words and 6964 distinct words, table 3 shows the summary statistics for this collection of texts.

Summary of statistics			
Madia	8.68796669	Kurtosis	140.484965
Standard error	0.33542491	Range	625
Median	2	Minimum	1
Mode	1	Maximum	626
Standard Deviation	27.9914047	Sum	60503
Sample variance	783.518735	Count	6964

Table 3 Summary statistics for pre-processed documents

The cut-off frequency was set to a value equal to the mean plus the standard deviation (in this case 8.69+28=37), i.e. a value equal to 37, leaving only 323 distinct words that meet this condition. Below are some of the results obtained in different conditions of the classification experiments carried out with the files shown in table 2.

First, a cross-validation was performed, which consists in giving a number n (n=10, in our case), dividing the data into n parts, and for each part, building the classifier with the remaining n-1 parts and testing the first part. The process is repeated for each of the n partitions. In our case, a stratified cross-validation is performed. We call it stratified when each of the parts retains the probabilities of the original sample (percentage of elements in each class). Table 4 shows the results obtained with the SVM and Naive-Bayes classifiers implemented. For each of them, the percentage of correctly classified instances is shown. We can observe that better results are obtained with SVM, in which when performing the experiment with stratified cross-validation we obtained 97.03 % of correctly classified instances, while with Naive-Bayes we obtain 96.12 %. In the next experiment we define a percentage with which the classifier will be built and the remaining part will be tested. We performed several experiments starting with only 10% for the creation of the classifier, then this percentage was increased to 30% and finally to 70%. In all cases results are presented using SVM and NB.

	VC (n=10)		CEP		% of instances in CEP
	SVM	NB	SVM	NB	
Accuracy	97.04	96.13	94.19	85.1	10
			96.43	96.1	30
			99.24	97.72	70

Table 4 Results obtained, cross-validation (CV) and with training and testing together (CEP)

ISSN: 2414-4835

Conclusions and future work

This paper presents a description of the automatic document classification activity. For the evaluation, a corpus of natural disasters consisting of newspaper articles in electronic format is used and is available upon request by email. With respect to the results obtained, we can observe that the number of correctly classified instances increases as more examples are used for the creation of the classifier. When we used only 10%, 94.19% of the instances were classified correctly, while with the classifier formed with 30% of the files, 96.42% of the instances were classified correctly, and finally with 70% of the files as part of the classifier, 99.24% of the instances were classified correctly. These results show the relevance and feasibility of the proposed methodology.

References

[1] Kjersti A. y Line E., Text Categorization: A survey, Norwegian computing Center, 1999.

[2] Sebastiani F., A Tutorial on Automated Text Categorization, Istituto di elaborazione dell'Informazione, 1999.

[3] Ayllón Lafuente, L. (2020). Evaluación de procesos de reconocimiento óptico de caracteres y detección de tablas para la clasificación automática de documentos y su integración en un gestor documental.

[4] Leiva, I. G., Ortuño, P. D., & Muñoz, J. V. R. (2019). Técnicas y usos en la clasificación automática de imágenes.

[5] Iglesias Hernández, G. (2020). Procesamiento automático de ilustraciones: Clasificación multi-etiqueta de cómics con Deep Learning.

[6] Hernández-Pajares, B., Pérez-Marín, D., & Frías-Martínez, V. (2020). Clasificación multiclase y visualización de quejas de organismos oficiales en twitter. TecnoLógicas, 23(47), 107-118.

[7] Hoz Maestre, J. A. D. L. (2020). Revisión exploratoria de literatura científica en acuicultura: Análisis de tendencias utilizando un mocelo probabilístico bayesiano y herramientas de machine learning.

[8] Guzmán Cabrera, R. (2019). Clasificación automática de opiniones en dominios cruzados. Computación y Sistemas, 23(4), 1541-1548.

[9] Rodríguez García, M. D. C. (2021). Utilización de Support Vector Machines para la Clasificación de Textos de acuerdo con los Objetivos de Desarrollo Sostenible.

[10] Navarro Clavería, C. F. (2020). Clasificación de patrones complejos de textura-color mediante extracción de características globales y locales, un clasificador SVM, y post-procesamiento.

[11] Lewis D. Evaluating text categorization. In Proceedings of the Speech and Natural Language Workshop, Asilomar, San Mateo, Cal, pp 312-318, 1991.

ISSN: 2414-4835