Article **Journal-Economic Development Technological Chance and Growth**
June 2022, Vol.8 No.10 33-39

Automatic document classification: the role of interclass similarity

Clasificación automática de documentos: el papel de la similitud entre clases

SORIANO-BURGOS, Claudio Isaac†*, LÓPEZ-RAMÍREZ, Misael and GUZMÁN-CABRERA, Rafael

Universidad de Guanajuato, División Ingenierías. Departamento de Estudios Multidisciplinarios. México.

ID 1st Author: *Claudio Issac, Soriano-Burgos* / **ORC ID:** 0000-0003-4937-6052, **CVU CONACYT ID:** 1150268

ID 1st Co-author: *Misael, López-Ramírez* / **ORC ID:** 0000-0003-0801-029X, **Researcher ID Thompson:** M-1412-2016, **Scopus ID Author:** 55981405600, **CVU CONACYT ID:** 386369

ID 2nd Co-author: *Rafael, Guzmán-Cabrera* / **ORC ID:** 0000-0002-9320-7021, **Researcher ID Thompson:** M-1412-2016, **Scopus ID Author:** 55981405600, **CVU CONACYT ID:** 88306

DOI: 10.35429/JEDT.2022.10.8.33.39 Received January 30, 2022; Accepted June 30, 2022

Abstract

The continuous increase of information in digital format requires new methods and techniques to access, collect and organize these volumes of textual information. One of the most widely used techniques to organize information is the automatic classification of documents. Automatic text classification systems have a low efficiency when the classes are very similar, i.e. there is overlap between them, and in this case it is very important to be able to identify those attributes that allow us to separate one class from another. In this paper we present the relationship between overlap between classes and classification accuracy. A public corpus with four classes is used for the evaluation, and each class is further separated by positives and negatives. The results obtained from four subsets with different number of training instances are presented, for each case the similarity plots, the accuracy value and the confusion matrices obtained are presented. The results obtained are very illustrative and show that the higher the similarity between classes, the lower the classification accuracy.

Supervised learning, Similarity coefficient, Automatic classification of opinions, Automatic classification of opinions

Resumen

El incremento continuo de información en formato digital obliga a contar con nuevos métodos y técnicas para acceder, recopilar y organizar estos volúmenes de información textual. Una de las técnicas más utilizadas para organizar la información es la clasificación automática de documentos. Los sistemas de clasificación automáticos de textos tienen una baja eficiencia cuando las clases son muy parecidas, es decir existe traslape entre ellas, y en este caso es muy importante el poder identificar aquellos atributos que nos permiten separar una clase de otra. En este trabajo se presenta la relación que existe entre el traslape entre las clases y la precisión de clasificación. Para la evaluación se utiliza un corpus público con cuatro clases y cada clase además separados por positivos y negativos. Se presentan los resultados obtenidos de cuatro subconjuntos con diferente número de instancias de entrenamiento, para cada caso se presentan las gráficas de similitud, el valor de la precisión y las matrices de confusión obtenidas. Los resultados obtenidos son muy ilustrativos y permiten comprobar que, a mayor similitud entre las clases, menor precisión de clasificación.

Aprendizaje supervisado, Coeficiente de similitud, Clasificación automática de opiniones

Citation: SORIANO-BURGOS, Claudio Isaac, LÓPEZ-RAMÍREZ, Misael and GUZMÁN-CABRERA, Rafael. Automatic document classification: the role of interclass similarity. Journal-Economic Development Technological Chance and Growth. 2022. 6-10:33-39.

* Correspondence from the Author: (E-mail: ci.sorianoburgos@ugto.mx)
† Researcher contributing first author.

Introduction

Supervised machine learning studies are gaining more importance recently due to the availability of an increasing number of electronic documents from different resources. Text classification can be defined as the task of automatically categorising a group of documents into one or more predefined classes. Therefore, the main goal of text classification is to enable users to extract information from textual resources and to deal with processes such as retrieval, classification and machine learning techniques to classify different patterns. In the text classification technique, an indispensable task is the preprocessing stage, where the data must be prepared to perform the classification task. Having computational tools to perform preprocessing tasks such as tokenisation of a text or the generation of graphs to visualise the similarity of documents would facilitate the classification work in the stages after preprocessing.

The problem of measuring similarity between documents and classes is not new in the state of the art, to date it is still an open research problem. When reviewing the most relevant works in the area we find for example the one reported in [1] where a measurement of similarity between words is proposed using the Jaccard coefficient, implemented in the programming language Prolog, and they conclude that the Jaccard similarity coefficient is adequate enough to be used in the measurement of word similarity, but without taking into account the similarity between sentences, which increases the similarity ratio. On the other hand, in [2] they propose five association measures in information retrieval from any encyclopaedia with a large number of documents in electronic format, such as Encyclopaedia Britannica or Wikipedia, the association measures are: Dice, Jaccard, overlap, simple matching and cosine coefficient.

Concerning text classification, in [3] they mention that texts and documents are unstructured datasets and that feature extraction and pre-processing are crucial steps for text classification applications. As part of this pre-processing they mention that data must be cleaned to omit unnecessary characters and features, in addition to tokenisation and stop word removal.

In the study reported in [4] they indicate that the measurement process in text similarity can be divided into text distance and text representation. Text distance can be divided into length distance, distribution distance and semantic distance; text representation is divided into string-based text, corpus-based text, single semantic text, multi-semantic text and graphical structure-based representation. Within string-based text similarity are the Sorensen - Dice and Jaccard coefficients. They conclude that these methods take into account the actual meaning of the text, however, they cannot be adapted to different domains and languages. The goal of supervised machine learning techniques for automatic text classification [5], is to determine whether or not a given document belongs to the given category by looking at the words or terms in that category. Among the most commonly used techniques are Naive-Bayes [6] and K-nearest-neighbours [7, 8].

Classification is considered as a form of supervised learning where, from input data and a training stage, class assignments (labels) are generated based on patterns that are separable, with a quantifiable accuracy. A previous step for the training of a classifier system consists of the transformation and preparation of the data that will form part of the training set, made up of a set of features that will be used as input to the system. Among the classifiers that have reported the highest accuracy are Naive-Bayes and Support Vector Machines (SVM). The Naive-Bayes classifier is the simplest instance of probabilistic classifiers, based on Bayes' theorem. From an input set, the classifier calculates the conditional probability that an element of that set belongs to one category or another, by calculating the ratio of the number of times the event occurs to the number of possible cases. SVMs consist of algorithms that represent predictive models for classifying a set of features through a linear function known as a hyperplane. In their implementation, 3 steps can be observed: a) feature selection; b) training and testing; c) evaluation. According to [9], SVMs achieve balanced performance and high accuracies, being ideal for tackling a variety of classification problems.

ISSN-2524-2024

SORIANO-BURGOS, Claudio Isaac, LÓPEZ-RAMÍREZ, Misael and GUZMÁN-CABRERA, Rafael. Automatic document classification: the role of interclass similarity. Journal-Economic Development Technological Chance and Growth. 2022

For accuracy estimation of a classification system, it is necessary to build the classification model. Firstly, the training set is constructed in order to implement it in the model; and secondly, a test set is used for the evaluation of the proposed classification model. The ratio of the labelled cases in the test set to the result obtained by applying the model will be the classification accuracy percentage. This ratio represents the number of correct predictions made by the classification system.

Methodology

In order to obtain the training set for the classifier system and to calculate the similarity between classes, a free corpus, available on the web, is used, which is described below.

Dataset

The dataset used consists of a corpus of text files with the comments of users of the Amazon online shop, concerning their experiences in the purchase of books, dvd's, electronics and kitchen items, separated into 2 categories (positive and negative) according to whether the purchase experience was positive or negative for each of the mentioned classes. Table 1 shows that the corpus consists of 9004 text files and how they are distributed in the different classes.

Category	Positives	Negatives	Total
Books	1037	1463	2500
Dvd	1394	1390	2784
Electronics	1015	949	1964
Kitchen	923	833	1756
			9004

Table I Distribution of the files in the corpus

As can be seen in table 1, there are not the same number of files for all categories, nor between positive and negative.

This set was used to carry out the evaluation of the system developed in this work. The purpose is firstly to test the similarity between the classes and secondly to test the intuitive idea that the higher the similarity, the lower the classification accuracy between them. Several sets were used to carry out the evaluation. Table 2 shows the balanced ensembles that were formed to carry out the evaluation of the methodology proposed in this paper.

The C100 class consists of 100 files of the positive category and 100 of the negative category; the C200 class consists of 200 files of each type and so on.

Class	Positives	Negatives	Archives
C100	100	100	1 - 100
C200	200	200	101 - 200
C400	400	400	201 - 400
C800	800	800	401 - 800

Table 2 Distribution of files in classes

The purpose of this file distribution is to calculate the classification accuracy as the number of instances in each class increases and also to see what happens to classes that have overlap, i.e., are more similar.

File generation

A Python script was implemented in order to generate three files, as shown in figure 1: a file with extension .arff, to perform the classification calculations in the supervised learning software Weka; a text file called Report.txt, where the paths of the categories and the most frequent words extracted from the text files are indicated; and a text file called Dictionary, where all the words extracted from the text files are recorded along with their frequencies, as well as the total number of words extracted.

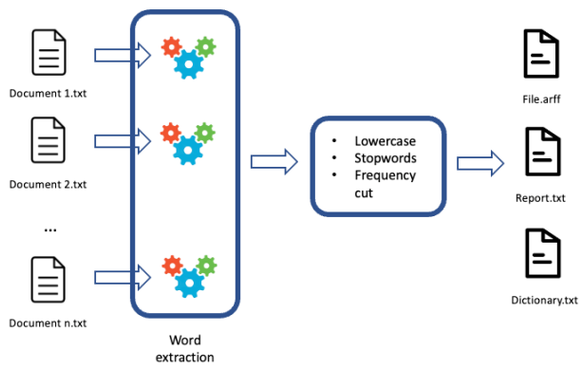

Figure 1 File generation

For the generation of the ARFF file, each file of each class is taken and the extraction of words is carried out, eliminating empty spaces and special characters, as well as converting all the words to lower case to be stored in a temporary list. Subsequently, when all the words in the files have been captured, the unique values and their frequencies are found.

This data is then stored in a dataframe where the process of stop words elimination and frequency cutting (both options if required by the user) will continue. The resulting dataframe is used for the generation of ARFF, Report and Dictionary files.

In order to increase the functionality of the developed algorithm, the function of loading the file Report.txt has also been implemented, with the purpose of being a source file for the modification or generation of new ARFF files, allowing to reduce the number of words and to perform new word searches in the files that make up the corpus in other paths.

Similarity measures

For the calculation of the similarity between documents and the elaboration of the similarity graphs, the Jaccard and Sorensen-Dice coefficients were calculated, which are described below.

Jaccard coefficient

The Jaccard coefficient is one of the techniques used to measure lexical similarity. It is a numerical value between 0 and 1, which measures the similarity between two text documents (1=completely the same; 0=completely different) considering the words that both documents have in common. This coefficient considers all elements or attributes to be equally important. It can be seen as a measure of similarity over sets, as shown in the following equation:

$$j(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

Sorensen-dice coefficient

The Sorensen-Dice coefficient is a similarity coefficient similar to the Jaccard coefficient which, likewise, its value is between 0 and 1, but unlike Jaccard, this coefficient considers the elements in common of a pair of sets to be of greater importance, as shown in the following equation:

$$s(A,B) = \frac{2|A \cap B|}{|A| + |B|}$$

These metrics are calculated in order to carry out the development of similarity plots, which are described in the following section.

Similarity plots

The general process that was implemented in the algorithm developed for the generation of the similarity plots is shown in figure 2. As can be seen, the Jaccard coefficient and the Dice coefficient were considered. It consists of taking each file and performing the tokenisation process, where the words that make up the text are separated, empty spaces and special characters are removed and converted to lowercase. These words are then stored in two lists to calculate the similarity coefficient and stored in a dataframe. When a file has been compared with the rest, a second file of the class is taken, and the process is repeated until all combinations are complete. The resulting data is used to make a scatter plot which will be the similarity plot for the selected class.

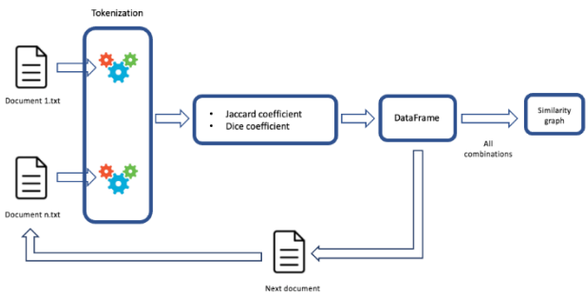

Figure 2 Generation of similarity plots

Results

For the evaluation of the developed algorithms, training and test sets taken from the positive feedback of each of the formed subsets (C200, C400, C600 and C800) were used. After tokenisation, the 1000 words with the highest frequencies were taken. The resulting training set was used with an 80-20 ratio (80% for training and 20% for testing).

Table 3 shows the results of performing the classification of these sets, using the previously described learning methods Naive Nayes (NB) and Support Vector Machines (SVM). We can see how in both cases the accuracy increases as the number of training instances increases.

Cross-validation		
Class	**NB**	**SVM**
C100	78.9	76.4
C200	78.1	73.7
C400	80.3	80.5
C800	81	86

Table 3 Cross validation results

ISSN-2524-2024

Similarly, table 4 presents the results obtained using the same learning methods, but now using the training and test set classification scenario. Unlike the previous scenario, in this case the training set is never seen by the test set, which makes it a more desirable scenario when there is a sufficiently large number of instances to be able to achieve efficient performance of an automatic classification system.

Training and testing		
Class	**Bayes**	**SVM**
C100	79.3	73.1
C200	74.3	77.3
C400	79.6	82.6
C800	81.5	86.3

Table 4 Training and testing

It can be observed, for most cases, that accuracy increases with increasing number of training instances. Now, it is time to ask what happens with the overlap between the classes, for which we will make use of the similarity plots for these same sets.

For the generation of the similarity plots, the Jaccard and Dice coefficients were calculated for the files with the positive comments contained in classes C100, C200, C400 and C800. The values of these coefficients are in the range of 0 to 1, where a value of 0 corresponds to a complete mismatch between the two files, while a value equal to 1 corresponds to the case where a file is compared with itself. According to the similarity plots generated, a higher value can be observed in the Dice coefficient, because this coefficient gives a higher importance to the elements in common, unlike the Jaccard coefficient, where all elements have the same importance. The graphs obtained for the sets under study are presented in figures 5 to 8.

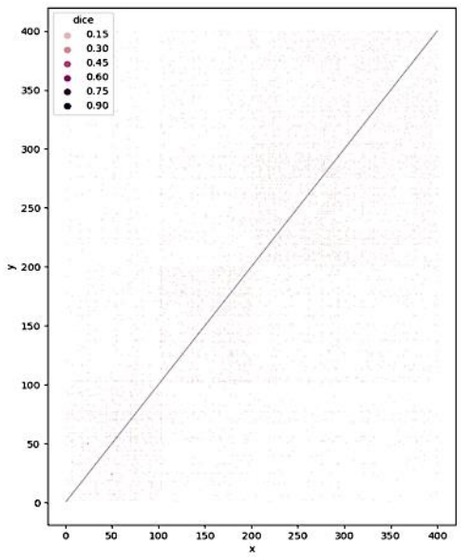

Figure 3 Similarity plots of the C100 class, positive

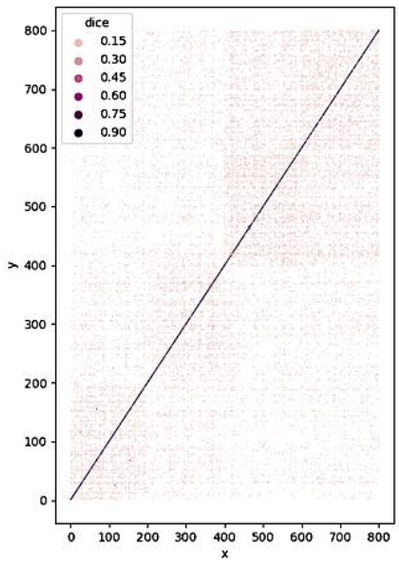

Figure 4 C200 class similarity plot, positive

In these graphs it can be seen that the number of points increases as the number of training instances also increases, this is because the vocabulary (number of words) is higher.

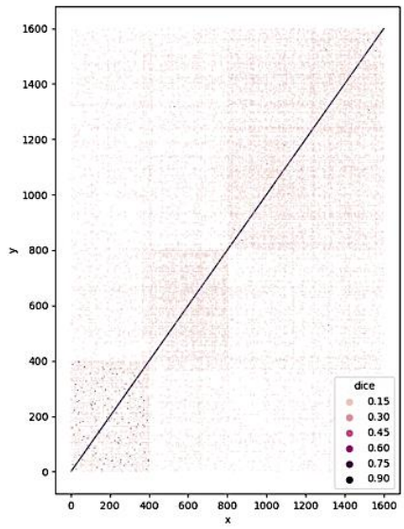

Figure 5 Similarity graph of class C400, positives

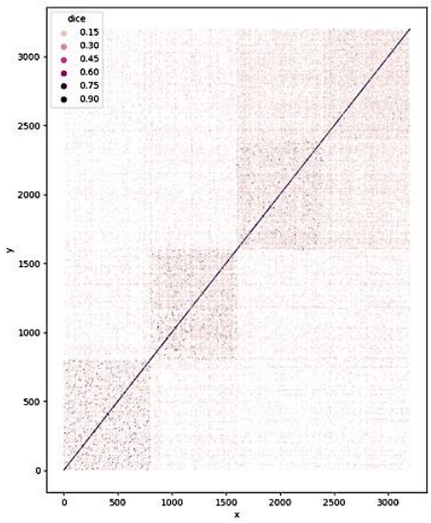

Figure 8 Class C800 similarity plot, positive

In the figures we can see how small tables are formed per class, this is because of the similarity between the language. For example, in figure 8, as described, there are 800 files for each class, that is 3200 files in total. We can observe a first frame from 0 to 799 (BOOKS), another from 800 to 1599 (DVD) and we observe between the last two classes (from files 1600 to 3199, corresponding to ELECTRONICS and KITCHEN) an overlap between these classes in all the similarity graphs, although in a more noticeable way from the set C200 onwards. Figure 9 shows the merging matrices for the set C200. The elements that are correctly classified are located on the main diagonal.

	Cross-validation				Training and testing			
	a	b	c	d	a	b	c	d
a=books	155	26	3	16	33	2	1	1
b=dvd	10	162	8	20	3	42	4	4
c=electronics	2	15	150	33	0	5	22	9
d=kitchen	2	15	32	151	0	2	21	21

Table 5 Confusion matrices for C200

The confusion matrices for the four sets under study and both classification scenarios are presented in Annex 1. We can see that, as expected, the classification system errs more between these two categories. For example, for Bayes, 33 files that were electronics were placed as kitchens and 32 that were kitchens were placed as electronics.

Conclusion

According to the results presented, it is observed that the classification accuracy increases as the number of training instances increases. Of the classification algorithms used, Support Vector Machines showed higher accuracy value as the number of instances increases. It can also be seen from the results obtained that as there is overlap between the classes this causes confusion in the classification method.

References

[1] S. Niwattanakul, J. Singthongchai, E. Naenudorn, S. Wanapu. Using of Jaccard Coefficient for Keywords Similarity. Proceedings of the International MultiConference of Engineers and Computer Scientists 2013 Vol I, IMECS 2013, March 13 - 15, 2013.

[2] S. Takale, S. Nandgaonkar. Measuring Semantic Similarity between Words Using Web Documents. International Journal of Advanced Computer Science and Applications, Vol. 1, No.4 October, 2010.

[3] K. Kowsari, K. Meimandi, M. Heidarysafa, S. Mendu, L. Barnes, D. Brown. Text Classification Algorithms: A Survey. Information 2019, 10, 2019

[4] J. Wang, Y. Dong. Measurement of Text Similarity: A Survey. Information 2020, 10, 2020.

[5] A. Kadhim. Survey on supervised machine learning techniques for automatic text classification. Artificial Intelligence Review. 2019

ISSN-2524-2024

[6] A. Mohammad, T. Alwanda, O. Al-Momani. Arabic Text Categorization Using Support vector machine, Naïve Bayes and Neural Network. GSTF Journal on Computing (JOC) Volume 5, Issue 1; pp. 108-115. 2016.

[7] S. Chen. K-Nearest Neighbor Algorithm Optimization in Text Categorization. IOP Conference Series: Earth and Environmental Science 108. 2018.

[8] M. Azam, T. Ahmed, F. Sabah, M. Hussain. Feature Extraction based Text Classification using K-Nearest Neighbor Algorithm. IJCSNS International Journal of Computer Science and Network Security, VOL.18 No.12. 2018.

[9] D. A. Pisner, D. M. Schnyer. Support vector machine. Machine Learning. 2020.

ISSN-2524-2024

SORIANO-BURGOS, Claudio Isaac, LÓPEZ-RAMÍREZ, Misael and GUZMÁN-CABRERA, Rafael. Automatic document classification: the role of interclass similarity. Journal-Economic Development Technological Chance and Growth. 2022