

Clustering Techniques for Document Classification

Julio Fernández¹, Jarvin A. Antón-Vargas¹, Yenny Villuendas-Rey^{1,2},
José F. Cabrera-Venegas¹, Yusbel Chávez¹, Amadeo J. Argüelles-Cruz³

¹ Universidad de Ciego de Ávila,
Departamento de Ciencias Informáticas, Ciego de Ávila,
Cuba

² Instituto Politécnico Nacional,
Centro de Innovación y Desarrollo Tecnológico en Cómputo, CDMX,
Mexico

³ Instituto Politécnico Nacional, Centro de Investigación en Computación, CDMX,
Mexico

janton@unica.cu, yenny.villuendas@gmail.com

Abstract. This paper is intended to study the existing classification and information retrieval techniques in order to use an algorithm that will group a set of documents. Therefore, the unfolding of knowledge in texts is selected as the proper methodology to be followed and the steps are explained in order to reach the unsupervised documents classification. After conducting an experiment with three of the most known methods of unsupervised documents classification and the assessment of the results with the Silhouette index, it could be observed that the better grouping was with four groups, whose main characteristic was to deal with subjects such as: information management information, systems management, artificial intelligence, and digital image processing.

Keywords: Document classification, clustering, silhouette.

1 Introduction

Within a few years Pc's have become a universal tool for all kinds of cultural, professional and commercial activities. Technological advancements in recent years have resulted in an exponential increase of digital information, which requires the development of specific tools for the retrieval and management of information [1]. That is the selection of a piece of information, among all the available information, for a specific user. This job is performed by the information retrieval systems [2, 3]. These systems deal with large databases composed of documents and use a model of representing information [4, 5], artificial intelligence techniques and data mining, they also process queries from users delivering the relevant documents in an appropriate range of time.

The main purpose of this kind of marketing is contacting a certain group of people to meet a specific goal (offer a service, insurance, credit card, etc.), but the problem of choosing the group of customers willing to buy the service is considered NP-hard [2]. Due to the existence of repositories in different institutions with an increasing amount of documents in digital format [6], it has become necessary to filter all the information in order to obtain the most accurate information needed [7]. That is, discarding all that is not of interest for the users and keeping what is useful. Sometimes it is interesting to know about a particular subject, but this would result in losing precious time looking for irrelevant information. In a large database, it is unthinkable to do a manual selection of these texts, i.e. it is very difficult to precisely know what all these texts are about. Therefore, it would be very useful to be provided with an automatic tool that would properly gather and manage text documents that meet some similar criteria to the search and explore the collection throughout the clusters obtained [7].

The rest of the paper is organized as follows. Section 2 details some materials and methods used and section 3 offers a discussion about the results obtained. Finally the paper ends with some conclusions and future research suggestions.

2 Materials and Methods

2.1 Collecting and Preprocessing Texts

This is the first step of the process and consists of extracting the plain information that appears in a set of documents that have been previously grouped [8]. Since all the theses from the faculty are in PDF format, it was necessary to find a mechanism to grab text from these files.

In order to extract text from PDF files, an expert library called PDFBox was used [9]. This library offers a wide range of preprocessing tasks such as text extraction, merging multiple documents into a single one, converting plain text into a PDF file, creating PDF files from images, printing documents and others. From all these features, it was decided to work with the extraction of text from a PDF file to plain text, where it will be easier to deal with.

In addition to this library, it was used another library called FontBox [9] that contains various types of fonts to make the PDFBox library fonts compatible with the most commonly known typefaces.

2.2 Lexical Analysis: Segmentation

Once the text from the documents have been obtained, the first operations to be performed on the text consist on segmenting large chains into corresponding words. This process is known as the separation of lexical components [10]. These tokens (which are just the words contained in the text) are obtained using the blank space characters for segmenting the whole text into independent words.

2.3 Filtering and Removing Stopwords

A second step is to filter all non-alphabetical characters such as numbers and punctuation marks, since they do not provide relevant information to the classification. Then, all the text is rewritten in lowercase, this will be useful to identify the same word, regardless it is uppercase or lowercase, be identified with the same word. Afterwards, another filtering is performed to eliminate those words that do not add relevant information such as pronouns, articles and conjunctions. These words are known as stopwords [11]. A list of Spanish and English stop-words was taken from [12], in order to eliminate those words from the search.

2.4 Standardization: Stemming

Once the stop-words are removed from the text, lexemes of the remaining words are sought in order to remove those words derived from the same stem. Words that share the same lexeme are treated as if they were the same word, this is especially useful for words that have different number and gender because they share the same meaning [13].

In order to find the lexemes from each word, a Java-based software was used. This software is Snowball [14], which is used in several areas of information retrieval and supports multiple languages including Spanish and English. An example of its functions appears in the table 1.

Table 1. Conversion from word to lexeme.

Words	Lexemes
<i>runs</i>	<i>run</i>
<i>taken</i>	<i>take</i>

A direct consequence of the use of the software is that it allows us to continue filtering the text because all those words and their variations that basically mean the same are suppressed. This affects nouns, adjectives, verbs and adverbs, but not conjunctions and prepositions because they were previously filtered as stop-words.

Table 2. Conversion to a unique word list.

Words	Stem
<i>take</i>	<i>take</i>
<i>taken</i>	
<i>runs</i>	<i>run</i>
<i>running</i>	

Notice that the words sharing the same lexeme are considered as the same word. Otherwise, it would be more difficult to find relationships among documents because words differing in just one letter would be considered different words. This would make

it difficult to accomplish if we take into consideration the variants of a same verb in different conjugation.

2.5 Unique Word List

To identify the set of documents, it must be created an alphabetically sorted word list having the words from all documents, the only requirement is that the same word should not be repeated. To remove repeated words, an alphabetically sorted word list will consider repeated words, and therefore, they will be removed from the list of words. To create a unique word list, the method used is to generate a list for each document with partial single words, that is, where there is neither repeated words or two or more words with the same lexeme or stop-words- later, the word list is alphabetically arranged.

After doing this with each of the documents a unique global word list is drawn up for all documents using the previously generated lists from partial words in each document. By making an individual process for each document, it is faster to create a list of unique words because in this process there has been many filtered words that provide a lot of extra processing.

2.6 Feature Generation

At the end of the previous section a basis of a vector space was obtained to represent each of the documents. It would be enough to take into account the times a word appears in a given document forming a vector with an equal length to the whole word list. This concept is often called term frequency (tf). Table 3 shows the representation of a textual corpus in the vector space [15], where the frequency of a term t in a document d is the sum of the number of times it appears in the document.

Table 3. Vector representation of a document corpus.

	Term ₁	Term ₂	...	Term _m
Document ₁	$tf_{d_1}(t_1)$	$tf_{d_1}(t_2)$...	$tf_{d_1}(t_m)$
Document ₂	$tf_{d_2}(t_1)$	$tf_{d_2}(t_2)$...	$tf_{d_2}(t_m)$
...
Document _n	$tf_{d_n}(t_1)$	$tf_{d_n}(t_2)$...	$tf_{d_n}(t_m)$

However, not all words are equally relevant to discriminate against among the documents since there are words that are very common to all documents and thus do not serve to distinguish a document from other.

Due to the previous vector representation for each document is modified so that those words that do not serve to distinguish between documents are not taken into account. For that it is applied the TF-IDF (Term Frequency–Inverse Document Frequency) which is defined in the following formula:

$$TF - IDF(t, d) = tf_d(t) * \frac{\log N}{df(t)} - 1, \quad (1)$$

where N is the total number of documents in the corpus, and df (document frequency) is the number of documents from the entire corpus in which that word appears. Thus, we see that if a word appears in all documents (such as Sp. "tener"), after this transformation its value in the table is null. The word count is performed using as a reference the unique word list that had been previously generated. An alphabetically arranged word list serves to look into each document and find the number of times each word is repeated in the text.

At the end of this text processing a matrix with a unique number associated with a corresponding associated number which will be used for further analysis and classification. This matrix is called word-documents matrix, and denoted by the letter M , it has a very large data and is based on the set of documents.

3 Discussion and Results

In this section the characteristics of the set of documents on which the experimentation is carried out are detailed. The experimental protocol is explained, describing the algorithms used to perform the clustering of the documents. Then is defined the evaluation metric to analyze the results of the experimentation.

3.1 Description of the Corpus of Documents

The FCI (Faculty of Computer Science) of UNICA (University of Ciego de Avila, Cuba) has a constantly increasing repository of theses in digital format. It has documents dating from the first graduation of computer engineering, class of 2006. These documents are in PDF format so it was sought to deal with this format.

Table 4. Description of the corpus used in this study.

Scholar year	Engineering Thesis	Master Thesis	Total
2005-2006	5	0	5
2006-2007	6	0	6
2007-2008	7	5	12
2008-2009	18	1	19
2009-2010	17	17	34
2010-2011	28	15	43
2011-2012	48	20	68
2012-2013	62	21	83
2013-2014	19	16	35
Total	210	95	305

PDF (Portable Document Format) is a document storage format developed by Adobe Systems. It is specially designed for documents that can be printed. This format is multi-platform since it can be viewed in all major operating systems (Windows, Unix \ Linux or Mac) without modifying either the appearance or the structure of the original

document. It also serves as the standard (ISO 19005-1: 2005) for electronic files containing documents intended to be preserved for a long term.

Since the academic year 2005-2006 to the 2013-2014, Computer Engineering at the FCI UNICA has stored over 235 theses, 210 are diploma papers, and 25 master theses (see Table 4).

3.2 Experimental Protocol

In order to find the best way to group documents diploma papers, a comparison is performed among the different algorithms for grouping documents: k-means, SOM and Hierarchical Agglomerative in its variants Single-Link, Complete-Link and Centroid.

The main disadvantage of these algorithms is that they require to set the initial number of groups to obtain in most applications, and in this particular case, there are no criteria to correctly specify this value. This is because the Corpus of Diploma Papers of Computer Engineering at UNICA is not labeled in groups.

To solve the problem of unknowing the number of groups to obtain, a necessary parameter to apply clustering algorithms. These algorithms were run in a range of 2 to a quarter of the number of documents to be grouped $\frac{N}{4}$. that is, a total of $\frac{N}{4} - 1$ runs were made for each algorithm.

Later, to determine the best grouping method, it was necessary to analyze the results with an index of internal validation. In accordance with several authors [16], one of the best performing indices in this regard is the Silhouette index.

The Silhouette index is an indicator of the ideal number of groups. A higher value of this index indicates a more desirable number of groups. Silhouette coefficient for a set is given as the average coefficient of each object silhouette sample, $s(i)$. This index can be used for both: a group of objects (cluster) or for each object. Silhouette coefficient for an object x is:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \quad (2)$$

Where $a(i)$ is the average distance from the object i to all other objects in their group and $b(i)$ is the average distance from the object i to all other objects in the nearest group. The value of $s(i)$ can be obtained by combining the values of $a(i)$ and $b(i)$ as shown below:

$$s(i) = \begin{cases} 1 - \frac{a(i)}{b(i)} & \text{if } a(i) < b(i) \\ 0 & \text{if } a(i) = b(i) \\ \frac{b(i)}{a(i)} - 1 & \text{if } a(i) > b(i) \end{cases} \quad (3)$$

According to the value of the total silhouette groups (structures) found they can be classified into:

- 0.71-1.0, the structures are solid.
- 0.51-0.70, the structures are reasonable.

- 0.26-0.50, the structures are weak and tend to be artificial alternate methods should for data analysis.
- <0.25, no structures are found

A value of $s(x)$ near zero indicates that the object x is on the border of two groups. On the contrary, if the value of $s(x)$ is negative, then the object should be assigned to the nearest group. This can be observed in Figure 1 with values forming silhouette 2 (b), 3 (c) and 4 (d) groups with the set of points of (a).

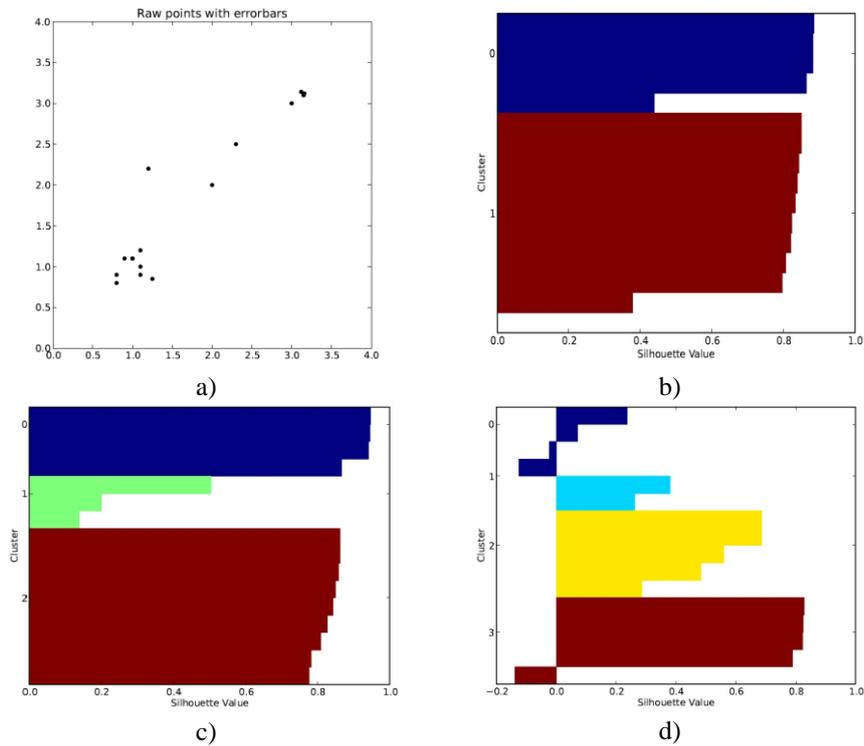


Fig. 1. Graphic representation of silhouette of different clusters.

As it can be seen, silhouette values are highlighted in the graphic with color values for different groups. A commonly used criterion for a better grouping is the average value of the outline of all objects in all groups. In this case, the greater Silhouette value will be chosen as the best grouping

3.3 Experiments and Discussion of the Results

In order to group documents in relation with their contents, the K-means, SOM and hierarchical Agglomerative algorithms are applied in combinations SingleLink, Complete-Link and Centroid on the data matrix characteristics obtained from the

Corpus of Diploma Papers. The input parameters using these algorithms are the set of data that is wanted to group. At the output of each algorithm a vector containing the labels is obtained with the group it belongs to each of the documents, as it can be seen in Figure 2.

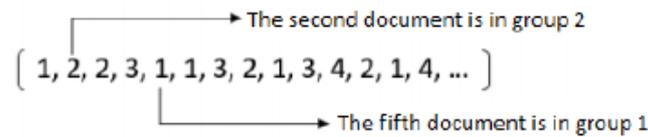


Fig. 2. Representation of algorithm’s output vector.

The total number of documents in the Corpus is 305. It was necessary to do 76 runs for each algorithm for a range of number of groups of 2 to 77. Table 5 shows the 10 best values of Silhouette for each algorithm and the number of groups obtained in each case.

Table 5. Clusters with different Silhouette value for each algorithm.

K-Means		SOM		Single		Complete		Centroid	
Index	Groups	Index	Groups	Index	Groups	Index	Groups	Index	Groups
0.7681	4	0.7114	3	0.7088	4	0.5287	2	0.7281	4
0.7082	3	0.5658	4	0.6699	3	0.5167	3	0.5540	3
0.7071	2	0.4826	5	0.5014	2	0.3232	4	0.4825	2
0.6511	73	0.4655	6	0.3069	5	0.1364	77	0.4259	6
0.6292	74	0.4268	7	0.2110	6	0.1260	76	0.3709	5
0.6271	75	0.3814	8	0.1820	75	0.1083	75	0.3078	7
0.6184	76	0.2728	12	0.1807	76	0.0993	74	0.2960	8
0.6111	77	0.2515	14	0.1741	74	0.0735	73	0.2876	9
0.6085	71	0.2304	9	0.1688	77	0.0663	72	0.2782	10
0.6047	72	0.2201	10	0.1603	73	0.0520	6	0.2511	11

As it can be seen in the table, the K-means algorithm has the highest value of the silhouette obtained (0.7681) forming 4 groups. It was followed by the hierarchical agglomerative Centroid-Link algorithm which also obtained 4 groups but with an average value of silhouette a little lower (0.7281). Thirdly, the hierarchical agglomerative Single-Link algorithm performs a grouping of 4 groups but also with an average silhouette (0.7088).

Likewise, if we plot the three best silhouette values for each algorithm on the number of clusters obtained it shows that the formed clusters are composed of 2, 3, 4 and 5 groups. Three, out of the five algorithms used in experimentation received the best value in silhouette for a cluster consisting of four groups. K-means algorithm is the best value obtained. These results can be seen in Figure 3.

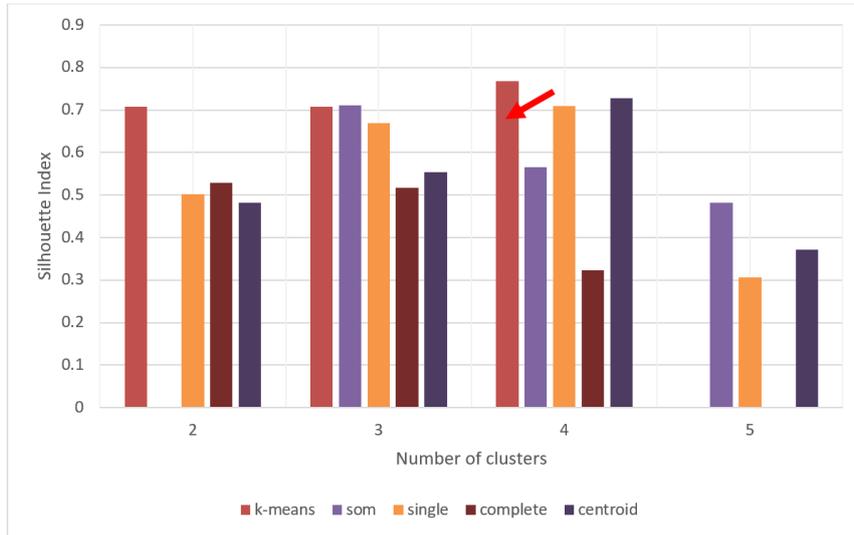


Fig. 3. Graphic of the three Silhouette values obtained by each algorithm with different number of clusters.

In order to understand more clearly the meaning of this Silhouette value, use Figure 4 where the silhouette of documents belonging to different groups can be seen, that is the documents belonging to the same group, appear together in a block.

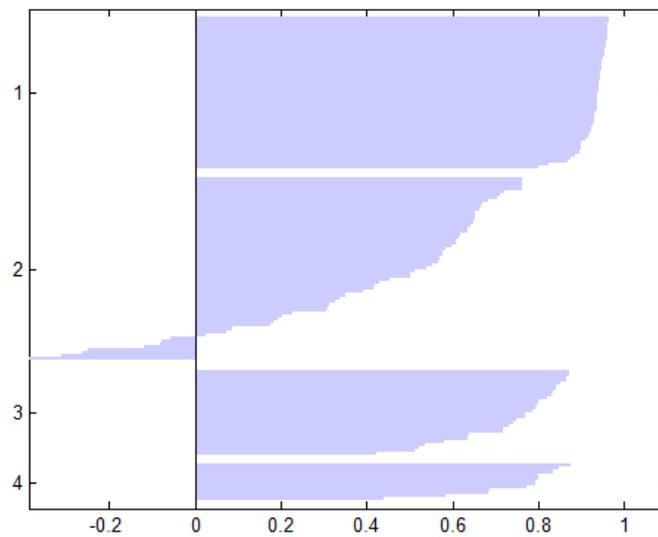


Fig. 4. Silhouette value of the documents by the four clusters obtained with K-means algorithm.

The silhouette value for each document is a distance that resembles how each document is similar to other documents within their own group. When compared with

the documents from other groups, taking values within a range of -1 to 1. As shown in Figure 4, the silhouette of the objects from the same group (for the four groups obtained) has close to 1 positive values and is wide which is an indicator of quality in the grouping. Only group 2 has a few objects with negative figures.

4 Conclusions

Clustering is amongst key text mining techniques for knowledge extraction from large collections of unlabeled documents. In this paper, we applied the Knowledge Discovery in Texts (KDT) methodology, and we use clustering to cluster a collection of thesis from the Faculty of Computer Science of the University of Ciego de Ávila in Cuba. Due to the lack of knowledge about the adequate number of desired clusters, we evaluated the different results according to an internal cluster validity index, which allow us to obtain a high-quality clustering. The best result corresponds to k-Means algorithm, with four clusters. The obtained clusters represent documents with different subjects, which are: information management systems, enterprise management systems, artificial intelligence and digital image processing.

Acknowledgments. The authors would like to thank the Instituto Politécnico Nacional (Secretaría Académica, Comisión de Operación y Fomento de Actividades Académicas, Secretaría de Investigación y Posgrado, Centro de Investigación en Computación, and Centro de Innovación y Desarrollo Tecnológico en Cómputo), the Consejo Nacional de Ciencia y Tecnología, and Sistema Nacional de Investigadores (SNI) for their economical support to develop this work.

References

1. Rodríguez, E., Berrocal, J., Paniagua, C., Rodríguez, Á.: Algunas Técnicas de Clasificación Automática de Documentos. Cuadernos de documentación multimedia 15, pp. 1–2 (2004)
2. Wang, C., Song, Y., El-Kishky, A., Roth, D., Zhang, M., Han, J.: Incorporating world knowledge to document clustering via heterogeneous information networks. pp. 1215–1224 (2015)
3. Castillo, J., Fernández, J.: Methodology of Preprocessing of documents for Systems of Recovery of Information (2008)
4. Chowdhury, Gobinda: Introduction to modern information retrieval (2010)
5. Bernotas, M., Karklius, K., Laurutis, R., Slotkiene, A.: The peculiarities of the text document representation, using ontology and tagging-based clustering technique. Information Technology and Control 36(2), pp. 217–220 (2015)
6. Gamare, P., Patil, G.: Web Document Clustering using Hybrid Approach in Data Mining. International Journal of Advent Technology 3(7), pp. 92–97 (2015)
7. Hassan, M., Karim, A., Kim, J., Jeon, M.: Cdim: Document clustering by discrimination information maximization. Information Sciences 316, pp. 87–106 (2015)
8. Mu, T., Goulermas, J., Korkontzelos, I., Ananiadou, S.: Descriptive document clustering via discriminant learning in a co-embedded space of multilevel similarities. Journal of the Association for Information Science and Technology 67(1), pp. 106–133 (2016)

9. Apache PDFBox | A Java PDF Library. <https://pdfbox.apache.org/> (September 2016)
10. Nalawade, Rahul, Samal, Akash, Avhad, Kiran: Improved Similarity Measure For Text Classification and Clustering. *International Research Journal of Engineering and Technology (IRJET)* (2016)
11. Wilbur, W., Sirotkin, K.: The automatic identification of stop words. *Journal of information science* 18(1), pp. 45–55 (1992)
12. Stopwords. <http://www.ranks.nl/stopwords>
13. Kanan, Tarek, Fox, Edward: Automated Arabic Text Classification with PStemmer, Machine Learning, and a Tailored News Article Taxonomy. *J. Assoc. Inf. Sci. Technol* (2016)
14. Porter, MF, Boulton, Richard: Snowball stemmer (2001)
15. Tang, Bo, He, Haibo, Baggenstoss, Paul, Kay, Steven: A Bayesian classification approach using class-specific features for text categorization. *IEEE Transactions on Knowledge and Data Engineering* 28(6), pp. 1602–1606 (2016)
16. Brun, Marcel, Sima, Chao, Hua, Jianping, Lowey, James, Carroll, Brent, Suh, Edward, Dougherty, Edward: Model-based evaluation of clustering validation measures. *Pattern recognition* 40(3), pp. 807–824 (2007)