ISSN 2816-8089

RESEARCH ARTICLE

Implementation of a Noise Filter for Grouping in Bibliographic Databases using Latent Semantic Indexing

Murilo Marques Armelin Gomes[1], **William Ferreira dos Anjos**[1], **Arun Kumar Jaiswal**[2], **Sandeep Tiwari**[2,5,6], **Preetam Ghosh**[3], **Debmalya Barh**[4,7], **Vasco Azevedo**[4], **Anderson Santos**[1*]

[1] *Department of Computer Science, Federal University of Uberlândia, MG 38400-902, Brazil.*
[2] *Postgraduate Program in Bioinformatics, Institute of Biological Sciences, Federal University of Minas Gerais (UFMG), Belo Horizonte 31270-901, MG, Brazil.*
[3] *Department of Computer Science, Virginia Commonwealth University, Richmond, VA-23284, USA.*
[4] *Department of Genetics, Ecology and Evolution, Institute of Biological Sciences, Federal University of Minas Gerais (UFMG), Belo Horizonte 31270-901, MG, Brazil.*
[5] *Post-Graduation Program in Microbiology, Institute of Biology, Federal University of Bahia, Salvador, BA, Brazil.*
[6] *Post-Graduation Programs in Immunology, Institute of Health Sciences, Federal University of Bahia, Salvador, BA, Brazil.*
[7] *Institute of Integrative Omics and Applied Biotechnology (IIOAB), Nonakuri, Purba Medinipur 721172, India.*

Abstract

Clustering algorithms can assist in scientific research by presenting themes related to some topics from which we can extract information more easily. However, it is common for many of these clusters to have documents that have no relevance to the topic of interest, thereby reducing the quality of the information. We can manage the reduced quality of information of clusters for a bibliographic database by dealing with noise in the semantic space that represents the relations between the grouped documents. In this work, we sustain the hypothesis of using the Latent Semantic Indexing (LSI) technique as an efficient instrument to reduce noise and promote better group quality. Using a database of 90 scientific publications from different areas, we preprocessed the documents by LSI and grouped them using six clustering algorithms. The results were significantly improved compared to our initial results that did not use LSI-based preprocessing. From the perspective of individual performance of

***Corresponding Author**: Anderson Santos, Department of Computer Science, Federal University of Uberlândia, MG 38400-902, Brazil; E-mail: santosardr@ufu.br*

***Received Date**: January 05, 2023, **Accepted Date**: January 24, 2023, **Published Date**: February 01, 2023*

__Citation__: Gomes MMA, dos Anjos WF, Jaiswal AK, et al. Implementation of a Noise Filter for Grouping in Bibliographic Databases using Latent Semantic Indexing. Int J Bioinfor Intell Comput. 2023;2(1):12-28.

the algorithms demonstrating the best results, CMeans was the one that got the highest average gain, with approximately 25%, followed by K-Means and SKmeans, with 17% each; PAM, with 16.5%; and EM, with 15%. The conclusion is that Latent Semantic Indexing has proven to be a helpful tool for noise reduction. We recommend its use to improve the cluster quality of bibliographic databases significantly.

Key Words: *SVD; LSI; Grouping; Dimensionality; Reduction*

1. Introduction

The amount of digital information stored increased from 2.6 exabytes, in 1986, to 295 exabytes in 2007 [1]—an increase of 113 times in 21 years. Along similar lines, research [2] points out that the internet grows exponentially, and its size doubles approximately every 5.32 years. Given this continuous and considerable growth of images, videos, documents, web pages, and other digital information, it is impossible to search, acquire, analyze and correlate this information manually and comprehensively. In this context, information retrieval is essential in proposing techniques and systems that enable the organization and efficient retrieval of large volumes of unstructured or semi-structured information.

Information retrieval systems are critical in searching for bibliographic references in scientific research. Using data from publications in the same area can reduce the time spent on experiments, provide better data sets, and prevent the researcher from developing work already done or identifying works whose results were surpassed by more recent publications. However, in some instances, this is not a trivial task if done manually, even with the help of internet search tools, due to many publications dealing with a specific topic. Consequently, considerable time is spent reading parts of the search results to address problems from the same domain. Such difficulty can be reduced if the user's search is returned in groups of semantically related documents, that is, documents that address the same problem domain, even if they use different words and expressions to describe it. One of the techniques that can achieve this goal is Latent Semantic Indexing (LSI) [3]. LSI is a way to improve the performance of search systems in the face of two known deficiencies: the use of synonyms in consultations, which reduces the scope of the results by limiting them to the searched term, and the polysemy, which causes the return of results not relevant to the search performed.

Indexing approaches determine whether a document is relevant to a query by seeking the occurrence of the terms of that query in the document. LSI considers a latent semantic space that truly represents the relationships between all terms and documents and seeks to obtain the best possible representation of this structure. Thus, those relations that were not observable become evident, allowing documents to be returned conceptually close to the search, even though the terms of that search do not occur in such documents [4].

2. Literature Review

2.1. Latent semantic index

Latent Semantic Indexing emerged as an alternative to the lexical comparison model of words. The main lexical comparison model disadvantages are the insensitivity to the variability of words that can be used to describe the same topic resulting in an incomplete collection of documents that could be returned from a search. Another is the insensitivity to words with more than one meaning, causing results not relevant to the search [4]. In the lexical model, a document is considered relevant only if it contains occurrences of the terms used in the consultation [4]. For example, in a search for "manioc," documents in which the occurrence of that word was not observed but contained the word "cassava" would not be returned. Search for the word "bank" would return all documents in which it occurred, even if part of them used it in the sense of an object and another part in the sense of a banking institution. To address this, LSI assumes that there is a latent semantic structure between terms and documents that can overshadow the phenomena mentioned above. When revealing such a structure, it would be possible for a search to return documents conceptually close to it [4].

LSI projects the matrix of document terms in a reduced dimensional space to approximate this semantic structure, making related terms and documents close together. Recovery is made from this space. Query terms are mapped to points in space, and the documents are returned based on their proximity to such points [3]. To obtain this space of reduced dimensions, LSI uses Singular Value Decomposition (SVD), a technique that originated in linear algebra in the 19th century [5].

2.2. Singular value decomposition

Singular Value Decomposition (SVD) is a matrix factorization technique in the field of linear algebra that emerged in the 19th century with the work of mathematicians Eugenio Beltrami (1835–1899), Camille Jordan (1838– 1921), James Joseph Sylvester (1814–1897), Erhard Schmidt (1876–959) and Hermann Weyl (1885–1955) [5].

In SVD, a matrix M can be represented as the product of matrices as in Equation 1,

$$M_{t\times d} = U_{t\times m}\Sigma_{m\times m}V^{T}_{m\times d} \qquad [1]$$

In equation 1, t is the number of rows in M, d is the number of columns, and m is the rank (the number of linearly independent rows or columns) determined by min (t, d). U_{txm} and V_{mxd} are matrices of orthonormal columns ($U^{T}U = I$ and $V^{T}V = I$), and Σ is a diagonal matrix of non-negative and decreasing singular values [6].

When removing the k small singular values of the matrix Σ_{mxm} and their corresponding columns in U and V^{T}, a new matrix is obtained,

$$M' = U_{t\times k}\Sigma_{k\times k}V^{T}_{d\times k} \qquad [2]$$

where M' is the matrix of rank k that best approximates M according to the least-squares method.

In practical terms, the matrix M' maintains only the most crucial k concepts, eliminating those considered noises, thus obtaining a compact representation of the original matrix without losing its main characteristics [7]. The reduced dimensional space of this matrix approximates the elements, making relationships that were not observable in the original matrix more evident [4].

The ideal k number of remaining elements in the matrix should be necessary and sufficient to eliminate noise and avoid the suppression of relevant information simultaneously. In practice, this value is usually defined empirically, and a parameter sweep on *k* can find an estimate that produces a satisfactory result [3].

2.3. Related works

A google scholar search for the words "identify," "similar," “LSI,” and "documents" returned nine thousand documents only for the last three years. We need resources beyond our capacity to investigate each of them, considering they are not clustered and filtered as proposed in this work. Because of that, we limited our text to cite just a few related to our work, some of them the newest and others considered more relevant to the keywords input.

Bradford tells us that using LSI in different areas after two decades contributed to ending several myths about LSI. For instance, classifications made by LSI are comparable to those made by humans or even better. Besides, LSI scales linearly with the data size [8]. Conclusions like these are essential in proposing using LSI to process the massive and growing data from the internet. Despite the enormous success of working with LSI, some researchers had different results. For instance, a tentative to identify topics covered in texts failed after using LSI over one thousand academic papers extracted from the areas of Biology, Medicine, Physics, and Social Sciences [9]. LSI was also successfully used to identify the customer preference for products and services provided by gas station companies in China [10]. Another example closer to our proposal is the use of LSI to classify documents for admin-case files of the Philippine National Police. In this work, documents were indexed based on file relationships and could return a search result as the retrieved information from files [11]. With the increasing size and the widespread use of XML schema and ontologies, it becomes tough to cope with large-scale schema matching. LSI also reaches encouraging results dealing with the XML schema matching problem [12].

Similarly, finding duplicate web pages is a vast and increasing challenge. For this purpose, LSI was successfully employed to detect conceptually similar documents, often not seen by textual-based identical detection techniques like Shingling and Simhash [13]. LSI was also applied to improve the grouping of different biological species. The groups formed to represent the evolutionary relationships between these species like that proposed by Lineu's taxonomy [14]. Genes' collection was used, and grouping was applied in two stages, one without the LSI and its use. The results were analyzed with a proposed metric proposed and demonstrated consonance with that taxonomy. Although the work also aims to highlight LSI's performance improvement, this study focused on the context of relationships between biological species, whose degree of similarity between elements is known and can be precisely measured. In another work, the authors use a collection containing documents from four different subjects. The full text of each document was extracted, and then certain words were removed based on a list of applied stop words in the preprocessing step [7]. This collection was then divided into training and test collections with the same number of documents. In the training collection, LSI was applied by varying the number of dimensions in an interval defined by the authors. Then, using the K-Means algorithm, each of the matrices obtained in the different dimensions was grouped. The dimension value that resulted in the best grouping performance for the training collection was directly applied to the grouping of the test collection. The results obtained by these tests could not demonstrate the technique's effectiveness in producing groups of greater relevance. Clustering solutions obtained with reduced dimensions performed inferior to the clustering solution in which no dimension reduction was applied. This paper investigates clusters' quality generated from LSI with a different approach from the one mentioned above, using only part of the text from the documents; preprocessing was applied based on the extraction of n-grams. In addition, the dimension reduction was carried out in the entire set of documents and submitted to a set of clustering algorithms.

This sample of related work comprises cases of success, failure, and enthusiastic results. The objective was to alert the reader that LSI is a powerful tool but is also capable of failing. Within the previous related experiments, we cannot ensure the authors correctly drew and executed the research with statistical rigor, even when they relate success. One should also pay attention to the LSI compliance with the proposed problem.

3. Methodology

3.1. Collection of documents

For the construction of the database, we collected 90 scientific publications, exclusively in English, divided between the scientific areas of economics and computer science. In both areas, we selected lines of research close to each other and with a variable number of elements. This approach aimed to facilitate the observation of the effectiveness of the LSI algorithm in separating opposing themes and joining themes known to be close or of a priori unobservable proximity. Among the 90 selected bibliographic references, 33 belonging to the Bioinformatics research line were provided directly by their authors, and 57 were obtained manually through consultations on the Google Academic search platform. Table 1 presents the selected lines of research, as well as the number of bibliographic references belonging to each one.

3.2. Normalization of the text

In the text normalization stage, we organize and treat the database to allow the latent semantic indexing algorithm to the set of documents, ensuring that the resulting distance matrix presents reliable results. Since the abstract sections present all the relevant information about the content of their respective research and, consequently, allow the comparison between different publications, we opted for the exclusive use of the abstracts to reduce the volume of text to be standardized and subsequently processed. The abstracts were manually extracted and transferred to a text file. We submitted this file to the normalization process to remove punctuation symbols and blanks, as little relevant information is extracted from them. In addition, since the algorithm was implemented to generate the distance matrix that distinguishes between uppercase and lowercase letters, all the letters were capitalized.

3.3. LSI application step

In the LSI application step, we submitted the normalized text file to the algorithm that produces the occurrence matrix, applied the LSI, and produced the resulting distance matrix for each dimension. For the construction and execution of this algorithm, we used the Scilab tool version 5.4.1. Scilab is a free and open-source numerical computing language with an extensive collection of functions from different mathematical areas, whose basic data structure is the matrix, making it easier to perform certain operations involved in the steps of the algorithm. To create the matrix of terms-documents, we used a preprocessing approach based on the extraction of n-grams.

Table 1: *Document collection categories and size.*

Category	Size
Computing - Data Mining	29
Marketing-oriented data mining	9
Preservation of privacy in data mining	9
Data mining on social networks	7
Data mining for counterterrorism	4
Computing - Bioinformatics	33
Genome sequencing	14
Analysis of genomes and related topics	19
Economy	28
Analysis of tourism in Latin America	7
Inequality in Latin America	5
Infrastructure in Latin America and Africa.	7
Inflation control in Brazil	9

N-grams are sequential or non-sequential selections of characters within a more extensive series, which can be n letters of a word (e.g., "Inflation" → "Infl") or n words in a sentence [15][16]. Its application was motivated by a study [15] instead of the traditional stemming model, a technique applying a set of rules to remove affixes from words reducing them to a basic form so that variations of the same word are classified as related concepts [17]. As the authors point out, the removal performed by stemming depends on sets of rules made specifically for the language in which the technique will be applied, a disadvantage that does not occur if n-grams are used.

We chose to extract a fixed number of n-grams per document to prevent the difference in size in the abstracts from affecting the result obtained by LSI. The chosen quantity, 300 n-grams, was the one that produced the best similarity between documents obtained in a trial-and-error process with different quantities of extractions being tested and compared with a model of document proximity that was empirically determined as ideal. With the matrix of terms-documents obtained, the algorithm applies a decomposition function by singular values native to the Scilab language, which decomposes the matrix into the submatrices U, Σ, and V^T on which a reduction k is applied, forming U_k, Σ_k, and $V_k{}^T$ (Equation 2). Finally, as the relationship between documents is of interest to the work, the $\Sigma_k.V_k{}^T$ matrix, which represents this relationship, is produced, and by calculating the Euclidean distance measure, the distance matrix is created and stored in a text file.

3.4. The grouping steps

In this step, we submit the distance matrices of each dimension obtained in the previous step to the grouping process. We selected clustering algorithms with different characteristics to provide further coverage to the study. It is possible to verify whether the LSI's expected effect can be observed for different grouping techniques.

The implementation of the algorithms selected for this function comes from two tools: Weka [18], which is open-source software that gathers techniques in machine learning and data mining, and the programming language R [19], in version 3.1.1, through its cluster and e1071 packages. Table 2 shows the algorithms used and their origin (tool).

Table 2: *Algorithms in this work.*

Algorithm	Tool
SimpleKMeans	Weka 3.6.11
WFP	R 3.1.1
HierarchicalClusterer	Weka 3.6.11
SimpleEM	Weka 3.6.11
SKmeans	R 3.1.1
CMeans	R 3.1.1

The first phase of tests aimed to identify, in each algorithm, the number of groups (k) that produced the best grouping solution from the original distance matrix with 90 dimensions. We measure the groupings' quality obtained with the variation of k in the range from 1 to 90. We used the results collected in this step as parameters for the second test phase. In the second stage, we conducted new tests to effectively verify this work's hypothesis's validity. In this step, we fix the number of groups to be created by each algorithm in the best result obtained in the previous step and, from this configuration, we repeat the grouping process for all the distance matrices obtained with the variation of dimensions (d value = {1, 2, ..., 89}).

Finally, we also opted for performing a complimentary test. We used only one algorithm with its number of groups fixed at 10, representing the number of classes of documents, and established a point of observation of the internal quality of the groups based on its contents.

3.5. Performance of used measures

To measure the performance gains and losses in the groupings resulting from the variation in the number of dimensions, we selected three metrics: Purity, Entropy, and F-Score, classified as external evaluation techniques, which are characterized by comparing, under some criteria, the group obtained to an ideal grouping [20]. We implemented the metrics mentioned in the Python language, version 2.7.6.

4. Results

Figure 1 shows the results of the relationship between the number of groups and performance obtained in each algorithm in an LSI free run. In this step, we quantify the results obtained exclusively using the F-Score metric since, among the set of metrics used in this work, it is the most suitable for the scenario in which there is variation in the number of groups.

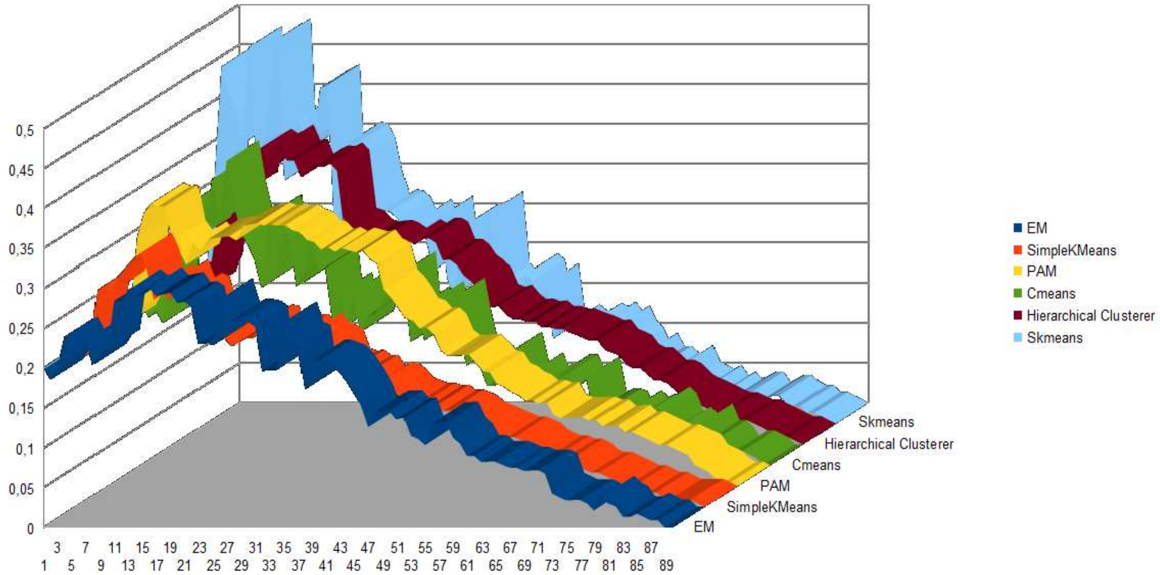

Figure 1: *The performance of the algorithms in the group variation test, created by Libre Office, according to the F-Score metric.*

We showed the best values for each algorithm (Figure 1) in Table 3.

Table 3: *Values of the best results for each clustering algorithm.*

Algorithm	K	F-Score
IN	16	0.302
SimpleKMeans	10	0.317
WFP	7	0.354
Cmeans	13	0.381
HierarchicalClusterer	16	0.375
Skmeans	11	0.483

The average result in Table 3 was 0.37, which is considered low since the F-Score metric varies between values from 0 to 1. Regarding the k values, which indicate the number of groups, they were in an interval relatively close to the number of classes in the database (10). We performed the second test varying the number of dimensions from 1 to 89 using the k values producing the best clusters. For each of these dimensions, we compare the quality obtained with the base result, which has no dimension reduction, to measure the percentage gain or loss of performance. In this test, in addition to the F-Score metric, we also use the metrics of Purity and Entropy to quantify the results under different quality parameters. The results measured according to the adopted metrics can be seen in Table 4.

Table 4: *Minimum, maximum, and average results with the application of LSI and the d value where the maximum (minimum for Entropy) was reached. The percentage gain or loss in quality is obtained when comparing the result without reducing an algorithm with its best value.*

Metric	Algorithm	Result without LSI	Minimum Result with LSI	Maximum result with LSI	Average with LSI	Best d value
Entropy	Skmeans	1.127	0.819	2.304	1.193	18
	Cmeans	1.487	1.137	2.259	1.448	21
	EM	1.314	1.040	2.053	1.372	9
	SimpleKMeans	1.69	1.454	2.371	1.818	3
	HierarchicalClusterer	1.124	1.097	2.247	1.428	87
	PAM	1.737	1.495	2.709	1.965	4
Purity	Skmeans	0.655	0.322	0.744	0.635	18
	Cmeans	0.522	0.344	0.633	0.543	21
	EM	0.611	0.368	0.655	0.565	9
	SimpleKMeans	0.468	0.333	0.544	0.468	19
	HierarchicalClusterer	0.622	0.333	0.633	0.555	87
	PAM	0.5	0.289	0.566	0.452	12
F-score	Skmeans	0.483	0.104	0.530	0.428	59
	Cmeans	0.381	0.120	0.499	0.335	13
	EM	0.302	0.089	0.356	0.264	18
	SimpleKMeans	0.317	0.131	0.383	0.269	21
	HierarchicalClusterer	0.375	0.090	0.377	0.264	87
	PAM	0.354	0.141	0.433	0.299	4

In Entropy, Figure 2 shows that the best averages repeated the ranges presented in Purity (Figure 3). The most significant gains in quality, in decreasing order, were: SKMeans - 27.3%; CMeans - 23.5%; MS - 20.8%; K-Means - 14%; PAM - 13.9% and HierarchicalClusterer – 2.4%.

The results measured according to the purity metric can be seen in Figure 3. We can observe three performance ranges: the first formed by the SKmeans algorithm, which obtained the highest average results, with 0.635; the second formed by the EM, HierarchicalClusterer, and CMeans algorithms, with averages 0.565, 0.555, and 0.543, respectively; and the third, formed by SimpleKMeans and PAM, averaging 0.467 and 0.452.

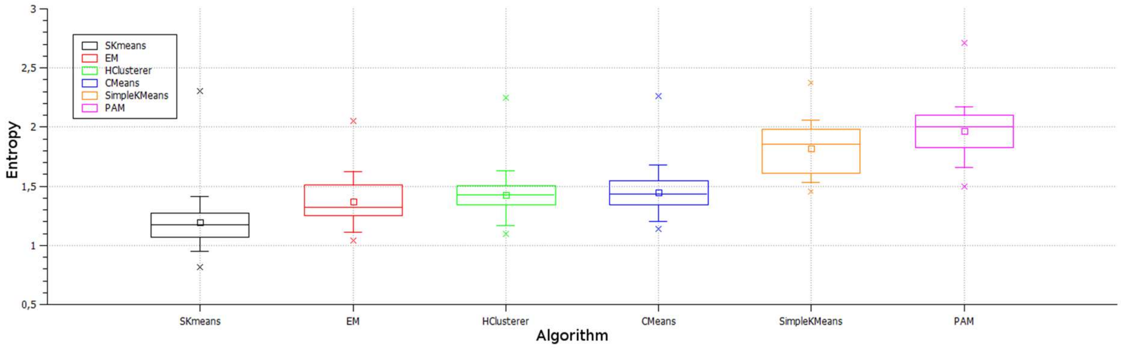

Figure 2: *A box chart of the results of the algorithms according to Entropy. Results closer to 0 indicate more significant quality gains.*

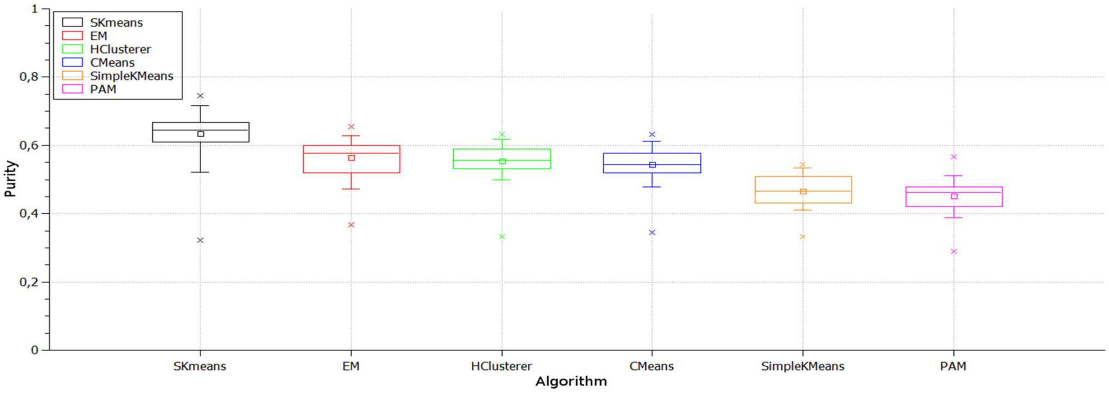

Figure 3: *A box chart of the results of the algorithms according to Purity.*

Comparing the maximum Purity and the values of the base grouping, shown in Table 4, the Cmeans and SimpleKMeans algorithms achieved the most significant gains with, respectively, 21.3% and 16.6%, followed by SKmeans, with 13.6%; PAM, with 13.3%; MS, with 7.3%; and HierarchicalClusterer, with 1.8%.

With the F-Score, represented in Figure 4, more excellent proximity to the mean of the algorithms is observed. SKmeans stand out with 0.428, followed by C-Means, with 0.335; PAM, 0.299; K-Means, 0.269; and HierarchicalClusterer and EM, 0.264. The tremendous quality gains were obtained in the following order: CMeans - 30.8%, PAM - 22.3%, K-Means - 20.6%, EM - 18%, Skmeans - 9.8%, and HierarchicalCluster – 0.3%.

We defined *k* as 10, equating to the number of classes in the set of bibliographic references. To simplify the analysis of the results, we used only SKmeans, which obtained the highest performance averages in the previous step. Again, we grouped by varying the dimensions from 1 to 89. The grouping solution obtained with 53 dimensions was chosen because Purity and F-Score pointed it out as the best result (Table 5). Although Entropy indicates that the reduction to 27 dimensions produced the best grouping, the result is only 0.2% higher than that obtained by the former, which is why it was neglected.

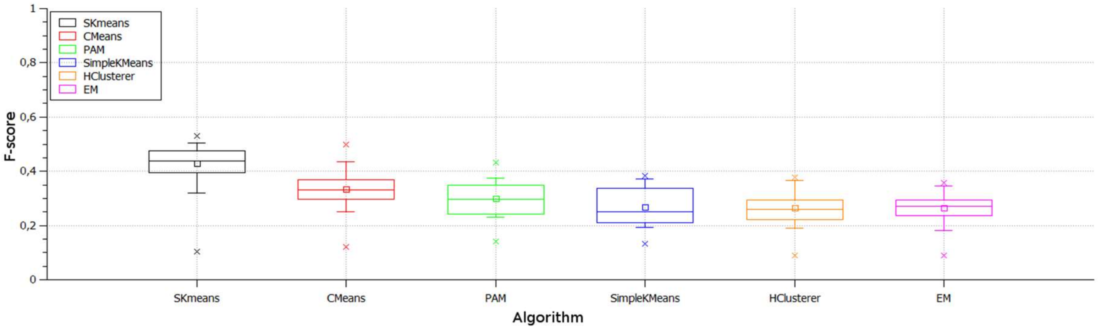

Figure 4: *Algorithms evaluation according to the F-Score.*

Table 5: *Base result (d value=90) and the best results obtained with reduced dimensions in the SKmeans algorithm, k=10.*

Dimensions	Purity	Entropy	F-score
27	0.733	0.872	0.517
53	0.755	0.874	0.536
90	0.544	1.434	0.415

Figures 5 and 6 show the distribution of elements in each base group and the group with 53 dimensions. For the base grouping, we noted that Group 1 was the only one whose elements belong to a single class, containing five out of nine publications on data mining aimed at marketing. In turn, groups 6, 7, 8, and 10 consist of publications from two or more classes associated with the same line of research or the same scientific area. Finally, in the remaining five groups, there is a co-occurrence between economics and computing publications.

With the reduction in dimensions, Figure 6 shows that the number of groups whose elements belong to a single class increased from one to two, being Group 6, formed by articles on infrastructure in Latin America and Brazil, and Group 7, with analyses on tourism in Latin America. It is also possible to observe that the number of groups containing publications on economics and computing decreased, remaining only in Group 2.

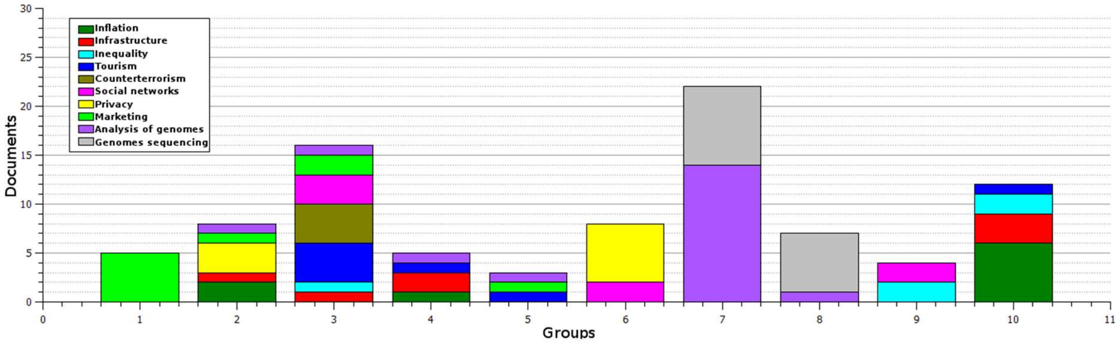

Figure 5: *The distribution of classes in the base group with no restrictions on the number of elements in the primary diagonal (d value).*

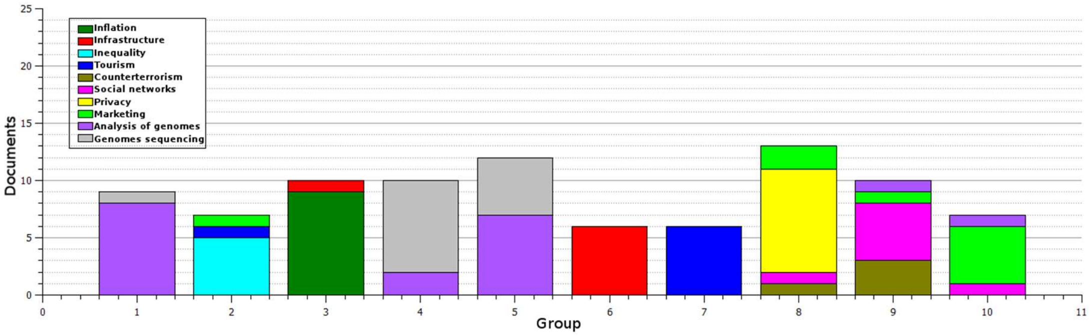

Figure 6: *Distribution of classes with d value = 53.*

In general, with the reduction to 53 dimensions, the average occurrence of 3.2 classes per group in the base grouping was reduced to 2.4. Groups' size discrepancy decreased, implying a reduction in the standard deviation of the number of elements per group from 6.0 to 2.45, as shown in Table 6.

Table 6: *Comparison of the distribution of elements by groups between the non-reduced grouping (d value = 90) and the grouping with 53 dimensions.*

Dimensions	Average classes per group	Smallest group size	Largest group size	Standard deviation
53	2.4	6	13	2.45
90	3.2	3	22	6.0

5. Discussions

When we start new research, the standard form of searching for literature background is searching for a couple of keywords on the internet. This method's main drawback resides in how the most widely used algorithms present the results: primarily listing the most relevant ones in order but not organizing the results in clusters. Conducting research, we used to look for ground articles offering the broader aspect possible, various research styles, and methods. After receiving from web searching sites hundreds of potential references to help with our research, we still should struggle with an endless list of articles, trying to figure out which are relevant for our citation. Clustering the result list could save us a lot of research effort; however, If the web searching sites could improve clustering by noise filtering methods, like LSI, better yet because we could have high-quality clusters saving us more time. We have given enough evidence in this work about the noise-filtering benefits of working with texts of different subject areas possessing common keywords. Also, the noise-filtering promoted using LSI is efficient since it works in a small portion of the original feature dataset, processing only the most prominent singular values.

Despite all we argue in previous sentences, one should remember there is no perfect solution for all kinds of problems. The LSI has benefits but also limitations. The central limit is to decide the better number of singular values we should use to acquire a better clustering, a still-opening problem. One can think of elaborate closed solutions based on fixed proportions of features in the set—for instance, some quarter of the number of elements under processing. However, there is no guarantee that such a strategy will always succeed.

Regarding the first test, in which we use the F-Score metric to identify the best base values for each algorithm, Figure 1 shows that in all algorithms, there is a pattern in which the performance decreases steadily as the number of groups increases to $k > 20$, reaching the value 0 when $k = 90$. This downward trend is explained by the spread of elements among more groups, which increases the occurrences of false negatives and decreases true positives, resulting in a sharp fall in the recall. Therefore, it is natural that the best k values for each algorithm were relatively close to the number of classes in the set of documents.

In the second test, observing Figures 2-4, and Table 4, the performance averages of the algorithms lead us to believe that the LSI produces results contrary to the expected. Except for CMeans, in the rest of the algorithms, such averages show that a good part of the range of quantity of dimensions generated negative results about the base grouping. Unexpected LSI performance occurred for higher quantities of dimensions and reduced dimensions to values less than 3. Such a phenomenon occurs because quantities of minimal dimensions suppress vital information from the data set. In contrast, high quantities model this set's noises and irrelevant characteristics to the detriment of the semantic structure [3]. Thus, the solution practiced by Deerwester [3]and other authors who use LSI is to find a positive value that produces satisfactory results and ignore those that do not.

In this sense, the initial doubt about the effectiveness of LSI is dissipated when only the maximum performance point of each algorithm is considered. From them, we found that all algorithms achieved superior results than their base groupings.

In general, analyzing the data in Table 4 of the second test, in the F-Score metric, the average value increased from 0.37, when there was no reduction, to 0.43, with the LSI representing an improvement of 17%. The same comparison in Entropy indicates that the average dropped from 1.41 to 1.17, repeating the 17% gain. Finally, the average purity value increased from 0.563 to 0.63, indicating a 12% gain. CMeans obtained the highest average gain, approximately

25%, followed by K-Means and SKmeans, 17%; PAM, 16.5%; MS, 15%; and HierarchicalClusterer, 1.5%.

The discrepancy in the result of the HierarchicalClusterer to the other algorithms does not mean inferiority. One hypothesis to explain the low gain concerns how the algorithm was configured in this work, using the complete connection criterion. Unlike algorithms considering averaging between points to determine the formation of a group, with the complete connection criterion, only the most distant points between two groups are considered. Reducing the dimensional space caused by LSI significantly affects the order of formation of the groups. The use of the medium binding criterion would bring better results.

In a previous work [7], Entropy, Purity, and F-Score pointed to an opposite effect to those noted in our second test. Objectively, the best cluster formed from the application of the LSI was approximately 9% lower than the base cluster. When comparing the two methodologies, it can be assumed that this low performance is associated with the authors' strategy of using a training subset of 59 documents to determine the best number of dimensions to be applied in the test subset. This quantity of documents may have needed to be increased for the projections in reduced dimensions to reveal the semantic structure of the 118 documents.

In complementary tests, to help understand the results, we classify the groups formed into three types: the first type, formed by groups that contain elements of a single class; the second type, formed by groups that contain elements from two or more classes associated with the same line of research or scientific area; and the third type, formed by groups that contain elements from different scientific areas. Comparing Figures 5 and 6, what became evident was the increase of the first and second types of groups in the grouping solution obtained with a consequent reduction of groups of the third type. While each group should represent precisely one class of publications, the increase in the number of groups of the second type can be considered positive since LSI identifies unknown semantic relationships between documents of the third type. While each group should represent precisely one class of publications, the increase in the number of groups of the second type can be considered positive since LSI identifies unknown semantic relationships between documents. Specifically, there should be a unique theme, the methodological similarity of tools, or any other characteristics that justify the joining of publications in these groups.

The reduction in the standard deviation of the distribution of elements by groups, from 6 to 2.45, shown in Table 6, reinforces that the LSI allowed the grouping process to capture more characteristics of the set of publications forming more specialized groups. Documents' distribution on Bioinformatics in Figure 5, mainly concentrated in group 7, which contains 22 of the 33 publications on the topic, and group 8, which contains seven elements, confirms a better LSI performance. After using LSI, Figure 6 shows that group 7 was disbanded. The Bioinformatics research line was redistributed into three groups, with 9, 10, and 12 publications, which was shown to be a positive effect by further specializing the groups on this line of research. Improvement's performance reinforces the results noticed in [7]. Although the authors have used specific metrics to evaluate these gains, the results also point to an increase in similarity between documents in the same group and groups with better-defined characteristics.

6. Conclusion and Future Works

In this work, we showed that Latent Semantic Indexing is an efficient tool to reduce noise in bibliographic databases submitted to a grouping process, providing groups that better represent the set of documents compared to groups generated without this noise removal. The experiments showed that all the algorithms had quality gains when the technique was applied. According to the external evaluation metrics used to quantify these gains, the spread of elements of the same class between different groups was reduced and the similarity of the obtained grouping solutions compared with an ideal hypothetical solution increased. The internal quality evaluation of one of the algorithms indicated that the groups formed with the LSI have more similar publications and better-defined characteristics. We also found that selecting appropriate dimension values was essential to achieve our results since part of these values produced adverse outcomes. It is possible to conclude that the Latent Semantic Indexing technique is an efficient instrument as a noise filter to realize groupings of bibliographic reference bases.

As a proposal to continue this work, we could create a graphical interface for users to load plenty of article abstracts to experiment with clustering after an LSI noise filtering. In this proposal, we could add one of our benchmark datasets to work as a positive control giving the user direction concerning chosen parameters. We used this technique previously with satisfactory results [14]. The ideal solution is for this process to occur directly in an internet search engine.

Funding

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

Authors' contributions

MMAG, WFDA, AS: conceived and designed the article; MMAG, WFDA, AS, AKJ, ST, PG, DB: collected data and wrote the article; AS, AKJ, ST, PG, DB, VA: provided critical review based valuable inputs.

References

1. Hilbert M, López P. The world's technological capacity to store, communicate, and compute information. Science. 2011;332:60-5.

2. Zhang GQ, Zhang GQ, Yang QF, et al. Evolution of the Internet and its cores. New J Phys. 2008;10:1-11.

3. Deerwester S, Dumais ST, Furnas GW, et al. Indexing by latent semantic analysis. J Am Soc Inf Sci. 1990;41:391-407.

4. Rosario B. Latent semantic indexing: an overview. Infosys 240, Spring. 2000.

5. Stewart GW. On the early history of the singular value decomposition. SIAM Rev. 1993;35:551-66.

6. Manning CD, Schütze H. Foundations of statistical natural language processing. MIT Press, Cambridge. 1999.

7. Antai R, Fox C, Kruschwitz U. The Use of latent semantic indexing to cluster documents into their subject areas. LTC 2011: 5th Language and Technology, Poznan, Poland. 2011.

8. Bradford R. Lessons learned from 20 years of implementing LSI applications. 2nd International Conference on Design of Experimental Search & Information Retrieval Systems, Padua, Italy. 2021.

9. Owa DLM. Identification of topics from scientific papers through topic modeling. Open J Appl Sci. 2021;11:541-8.

10. Watada J, Roy A, Vasant P. Preference identification based on big data mining for customer responsibility management. Int J Intell Technol Appl Stat. 2020;13:1-24.

11. Aquino AM, Chavez EP. Analysis on the use of Latent Semantic Indexing (LSI) for document classification and retrieval system of PNP files. 2nd International Conference on Material Engineering and Advanced Manufacturing Technology, Beijing, China. 2018.

12. Moawed S, Algergawy A, Sarhan A, et al. A latent semantic indexing-based approach to determine similar clusters in large-scale schema matching. Adv Intell Syst Comput. 2014;241:267-76.

13. Roul RK, Mittal S, Joshi P. Efficient approach for near duplicate document detection using textual and conceptual based techniques. In: Kumar Kundu MK, Mohapatra D, Konar A, et al. (eds) Advanced Computing, Networking and Informatics- Volume 1. Smart Innovation, Systems and Technologies, Springer, Cham. 2014.

14. Santos AR, Santos MA, Baumbach J, et al. A singular value decomposition approach for improved taxonomic classification of biological sequences. BMC Genom. 2011;12:S11.

15. Mayfield J, McNamee P. Single N-gram Stemming. Proceedings of the 26th annual international ACM SIGIR Conference on Research and Development in Information Retrieval, Toronto, Canada. 2003.

16. Cavnar WB, Trenkle JM. N-Gram based text categorization. Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval, Las Vegas, USA. 1994.

17. Lovins JB. Development of a stemming algorithm. Defense Technical Information Center, Virginia. 1968.

18. Hall M, Frank E, Holmes G, et al. The WEKA data mining software. ACM SIGKDD Explorations Newsletter. 2009;11:10-8.

19. R Core Team. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria. 2019.

20. Manning CD, Raghavan P, Schutze H. Introduction to information retrieval. Cambridge University Press, UK. 2008.