An Apache Spark Implementation for Text Document Clustering

Elias Dritsas*, Maria Trigka*, Gerasimos Vonitsanos†, Andreas Kanavos‡ and Phivos Mylonas‡
*Department of Electrical and Computer Engineering
University of Patras, Patras, Greece
{dritsase, trigka}@ceid.upatras.gr
†Computer Engineering and Informatics Department
University of Patras, Patras, Greece
mvonitsanos@ceid.upatras.gr
‡Department of Informatics
Ionian University, Corfu, Greece
{akanavos, fmylonas}@ionio.gr

***Abstract*—As the volume of data generated and stored on a daily basis is constantly increasing, the need for finding techniques in terms of the automated discovery of information from them has arisen. This purpose can be effectively solved with the use of text mining, which uses methods derived from data mining, information retrieval, machine learning, as well as natural language processing. This paper addresses the problem of extracting textual information from large collections of documents by efficiently exploiting clustering techniques in a cloud computing infrastructure. The clustering was performed using three different algorithms, namely k-Means, Bisecting k-Means, and Gaussian Mixture Model (GMM). To evaluate the quality of these methods, we experimented in the Apache Spark distributed environment, on several well-known datasets, the documents of which have been manually clustered.**

***Index Terms*—Text Mining, Cluster-based Methods, Clustering, Apache Spark, Knowledge Extraction**

I. Introduction

The volume of data produced, stored and used daily is constantly raising. The question that arises from this fact is the effective utilization of the available data for knowledge extraction, which is useful in the real world; data mining process serves this purpose. Specifically, it uses clustering or classification algorithms, along with principles of statistics, artificial intelligence, machine learning, and dataset systems for extracting potentially useful information and standards. Classification is perhaps the most well-known and popular technique of data mining and refers to the process of predicting a highlighted class for an unmarked item. It is essentially a division of objects so that each object could correspond to one of several mutually exhaustive and exclusive categories, known as classes [21].

On the other hand, clustering is a technique similar to classification, with the main difference being that in this case, the datasets, called clusters, are not predefined but are then derived from the data. A cluster is a collection of data objects intertwined within a cluster and non-clustered with objects in other clusters. Similarity measures are used for predefined features to determine the clusters generated, depending on the data. Clustering is also well known as non-supervised learning [30].

The data collection method, known as document clustering, combines features from machine learning, natural language processing, and information retrieval [4]. A specified similarity measure, divides documents into various clusters where the documents in each cluster share common properties. Users require support from efficient and high-quality document clustering algorithms to efficiently navigate, summarize, and organize the information. Hard clustering and soft clustering consist of the two main sub-categories of document clustering. The strict mapping of a document to a cluster is calculated via hard clustering, meaning that each document is mapped to the same cluster by a collection of unique clusters. Each document may appear in as many clusters as desired when using soft clustering; thus, a set of overlapping clusters is created [1].

Given the fact that there are several clustering algorithms, their categorization is difficult since many have characteristics belonging to more than one category. However, the most crucial clustering methods are divided into the following basic categories: partitioning, density-based, hierarchical, and lattice algorithms. Classical clustering algorithms may be inappropriate when applied to large and dynamic datasets. This is due to their high complexity (usually $O(N^2)$) and the notion that there is sufficient memory to store the data during the algorithm execution and that all data simultaneously exist. However, previous assumptions are unrealistic when clustering algorithms are required to process big data [5].

The rapid increase of internet users in recent years, combined with the vast amount of information, has created the need for services and tools to provide and process reliably this information. This purpose is served by cloud computing as a shared pool of computer resources (networks, servers, storage, applications, services, etc.) can be immediately fed and interacted with using the cloud computing concept, which enables on-demand internet access to the resources [26].

978-1-6654-8727-6/22/$31.00 © 2022 IEEE

One of the most widely used frameworks for cloud-based data analytics applications is Apache Spark [6], [23]. It is a powerful open-source Hadoop computing platform which is written in the Scala programming language. As Spark is a general-purpose infrastructure for computer clusters, it is mainly used for various applications. Furthermore, the development of cloud computing infrastructures has led to the creation of new techniques and algorithms for data processing and analysis, as well as machine learning libraries aimed at exploiting its potential [28].

From the machine learning point of view, Spark provides its users with many traditional techniques, such as classification, clustering, regression, which constitute the so-called MLlib library [19]. In our paper, we aim to efficiently and effectively apply document clustering to extract meaningful textual information. For this purpose, the Apache Spark framework was used as it is characterized by high execution speeds, tasks parallelization and system cache memory utilization when implementing different algorithms. For the evaluation of the clustering quality, we measured the F-measure metric, which is the harmonic mean of the Precision and Recall metrics along with the execution time of each algorithm.

The rest of the paper is organized as follows. Section II describes the relevant to the subject works. Besides, Section III analyzes the methodology followed, whereas in Section IV, the acquired research results are captured. Finally, discussion and conclusions are outlined in Section V.

II. Related work

Document clustering methods partition a collection of documents into cohesive and distinct clusters based on content similarity. These techniques often use features represented as word embeddings or Term Frequency-Inverse Document Frequency (Tf-Idf) matrices.

Document and query processing has gained much amount of attraction in recent years [13]. Specifically, information processing focuses on information redundancy [2], [3]. These techniques leverage greedy algorithms, like Maximum k-Intersection and Maximum k-Union in order to allocate the maximum intersection of similar information between pairs of documents. Upon locating the same context, a new document is being created, which contains this specific context along with any new information contained in texts participated in the intersection process over a specified threshold. The newly created content is checked with the coherence metric that concludes whether the derived text is logical and valid.

More to the point, in a series of additional papers [14], [15], [16], [27], a number of interesting approaches are being explored in order to provide further information in the text as well as terms annotation. The process of Wikification utilizes the structure of Wikipedia pages so as to assign additional weights to the scores assigned to terms through WordNet's disambiguation process. Moreover, authors in [9] used the TAGME technique in conjunction with WordNet results, and the final score was assigned to the texts before being returned for a specific query.

In following, this section presents a number of relevant works in terms of clustering methods applied to a text document. These papers either enhance previously established methods in terms of their clustering accuracy or evaluate their performance in distributed environments focusing on big data.

A. Clustering Perspectives

Text document clustering is applied using the Spectral Clustering algorithm with Particle Swarm Optimization (PSO) [11]. In terms of clustering accuracy, the suggested approach surpasses Spherical k-Means, traditional PSO as well as Expectation-Maximization (EM) algorithm [20].

Additionally, authors in [10] devised a technique that specifies the number of clusters by initially calculating a threshold value that serves as the k-Means center. Two data points will be included in the same group in a given iteration of k-Means if the Euclidean distance between them is less than or equal to the threshold value. If the suggested method is not applied, a new cluster with different data points will be constructed.

Furthermore, different features are retrieved from data, such as sentence-level and phrase-level embeddings, Tf-Idf features, and document embeddings. The data is then clustered using three different techniques (k-Means, Bisecting k-Means, and Affinity Propagation) [22]. According to the results, Tf-Idf features with use of k-Means clustering algorithm outperformed all other clustering methods.

B. Clustering in Distributed Environments

The efficiency of MapReduce in document clustering using a parallel k-Means algorithm is measured in [29]. To validate the superiority of their clustering approach, authors compared it with the sequential k-Means in varying sizes of document datasets.

A more recent work improves the efficiency of the k-Means algorithm and its time complexity by utilizing the Hadoop cloud-based platform, a distributed database appropriate to simulate the shared memory space and parallelize the algorithm [18]. In [24], k-Means is compared with a probabilistic clustering method, namely the Gaussian Mixture Models, in a cloud computing paradigm. This method has higher computation time, but it can identify more complex patterns achieving better clustering with distinct boundaries.

III. Methodology

The main steps of the elaborated framework are illustrated in Figure 1. Pre-processing, $Tf - Idf$ feature extraction, and clustering techniques comprise the process three main components.

A. Pre-processing

Initially, each input document is converted to its alphanumeric form based on punctuation marks, spaces, and new lines. Its contents are scanned and in following are divided into an alphanumeric table according to a set of separating symbols; this process is called verbal analysis. Many of these alphanumerics can be HTML tags as the texts in the collection

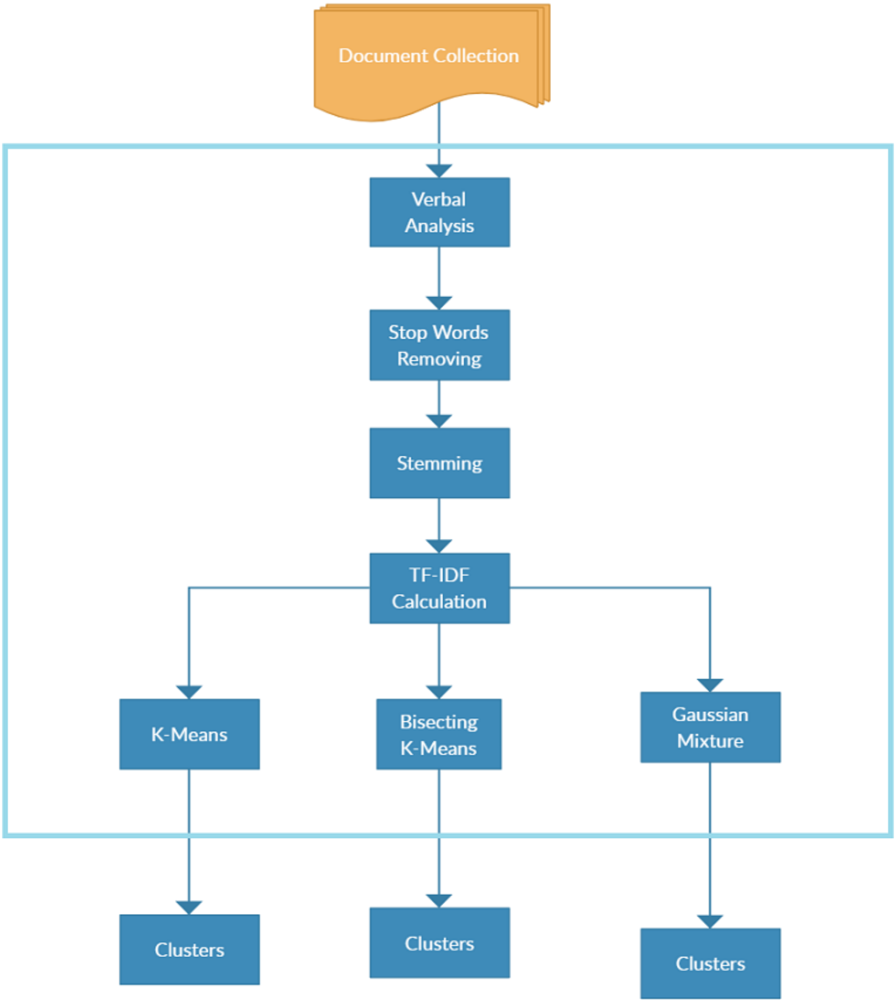

Fig. 1. Text Document Clustering Framework

may be retrieved from the World Wide Web in HTML format. These HTML tags can provide helpful information as they can distinguish the title of a web page as well as metadata, headings, etc. However, these tags are removed since they are irrelevant to the each document's content [7].

In the next phase, stop words are removed so that the updated form will contain only terms that can be related to the content of each document. Finally, the result of this process is given as input to the stemming algorithm for converting all the document terms to their root format and all characters are converted to lowercase. So, for example, "playing", "played", "plays" must be reduced to its common root, "play".

B. *Tf-Idf*

The next component of the methodology constitutes the vector representation of the preprocessed documents using the $Tf - Idf$ method [17]. We assume a corpus $\mathcal{D} = \{d_1, d_2, \ldots, d_N\}$ of N documents where t is a word term in the documents collection. The Term Frequency Tf counts how many times the word t, which is defined in a dictionary, occurs in a document and is computed as:

$$Tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}}, \tag{1}$$

where $n_{i,j}$ represents the number of occurrences of t_i in document d_j, $\sum_k n_{k,j}$ represents the total number of occurrences of words in document d_j and K is the number of words.

Besides Tf, the Document Frequency DF captures the number of times each word appears in the collection $\mathcal{D}$ and is calculated as:

$$DF_{i,j} = \frac{|d_j \in \mathcal{D} : t_j \in d_j|}{|\mathcal{D}|} \tag{2}$$

where $|d_j \in \mathcal{D} : t_j \in d_j|$ represents the number of documents that word t_j occurs and $|\mathcal{D}|$ is the total number of documents.

The IDF, which is the inverse of DF, measures the importance of words in the collection of documents and is defined as:

$$IDF_{i,j} = \log\left(\frac{|\mathcal{D}|}{|d_j \in \mathcal{D} : t_j \in d_j|}\right) \tag{3}$$

Based on the above, $Tf - Idf$ is computed as:

$$Tf - Idf(t, d) = Tf(t, d) \times IDF(t) \tag{4}$$

C. Clustering

Three different clustering algorithms including k-Means, Bisecting k-Means, and Gaussian Mixture Model (GMM) were utilized and tested in the present work, implemented with use of the Apache Spark Machine Learning Library (MLlib). Specifically:

- **k-Means** algorithm [8], [25] has been widely used in data mining and information retrieval. This grouping aims to separate N documents into groups so that each document belongs to the cluster with the closest means, which serves as a typical sample of the cluster. The algorithm's input is the $Tf - Idf$ scores of all documents along with the desired number of k clusters. The documents clustering with use of $Tf - Idf$ value leads to groups of similar documents according to the importance of their words. Based on the Euclidean distance between the $Tf - Idf$ value of the document and a center value of each cluster, k-Means clustering algorithm determines the center of the cluster that represents a group of documents with a particular subject and assigns a document to a cluster with a high degree of similarity [17].
- **Bisecting k-Means** [22] is a variation of k-Means and is analyzed in the following steps:
  1) Applies random initialization of the centroids.
  2) Selects a cluster.
  3) Applies traditional k-Means algorithm on the cluster setting where the number of clusters is $k = 2$.
  4) Repeat step (3) for the number of iterations specified and selects the partitioning with the lowest Sum of Squared Errors (SSE).
  5) Steps (2), (3) and (4) are repeated until the number of desired clusters are obtained.
- **Gaussian Mixture Models** [24] is an unsupervised clustering technique that forms clusters based on probability density estimations using the Expectation-Maximization algorithm [12]. A GMM groups the data points using the soft clustering approach for distributing the points in different clusters. Each cluster is modelled as a Gaussian distribution.

IV. Experimental Evaluation

In this section, our aim is to evaluate the aforementioned clustering techniques in a distributed environment. Specifically, this work was implemented and tested in Apache Spark on a 2.2GHz, 4-core computer system, with 8GB RAM, and a Unix operating system. The evaluation was implemented in terms of two aspects, clustering accuracy and time efficiency, under two different collections of documents. The F-measure of cluster j and class i was used to evaluate the clustering quality, which is the harmonic mean of precision and recall. Following [31], this metric is defined as:

$$F - Measure(i, j) = 2 \cdot \frac{Precision(i, j) \cdot Recall(i, j)}{Precision(i, j) + Recall(i, j)} \tag{5}$$

where Precision(i,j) measures the ratio of retrieved documents from j-th cluster that belong to the i-th class while Recall(i,j) measures the ratio of retrieved documents from the i-th class that belong to the j-th cluster.

To evaluate the clustering methods, the Webkb[1] dataset was utilized, which has been manually categorized into seven categories. The dataset includes 5163 websites from Computer Science departments and is divided into the following categories: student, staff, faculty, department, course, project, and other. In Table I, the F-measure values per category and clustering method, are recorded. We notice that student and other achieve the highest F-measure with course having the lowest metric, whereas Gaussian Mixture Model outperforms the other two clustering techniques.

TABLE I
WEBKB DATASET: F-MEASURE EVALUATION IN TERMS OF CLUSTERING METHODS

Category	k-Means	Bisecting k-Means	Gaussian Mixture Model
student	0.75	0.75	0.77
staff	0.61	0.61	0.63
faculty	0.66	0.66	0.71
department	0.71	0.71	0.74
course	0.58	0.58	0.66
project	0.69	0.69	0.69
other	0.79	0.79	0.70

The 20-Newsgroups[2] dataset, which contains 18846 documents, was also used for testing our techniques. Some forums are very closely linked (e.g. comp.sys.ibm.pc.hardware and comp.sys.mac.hardware), while others are not relevant (e.g. misc.forsale and soc.religion.christian). Table II depicts the categories of the collection and the number of documents in each category are captured.

TABLE II
20-NEWSGROUPS DATASET: CATEGORIES AND NUMBER OF DOCUMENTS PER CATEGORY

Category	Records	Category	Records
alt.atheism	799	rec.sport.hockey	999
comp.graphics	973	sci.crypt	991
comp.os.ms-windows.misc	966	sci.electronics	984
comp.sys.ibm.pc.hardware	982	sci.med	990
comp.sys.mac.hardware	963	sci.space	987
comp.windows.x	985	soc.religion.christian	996
misc.forsale	975	talk.politics.guns	909
rec.autos	989	talk.politics.mideast	940
rec.motorcycles	996	talk.politics.misc	775
rec.sport.baseball	994	talk.religion.misc	628

Table III presents the values of the F-measure calculated for the 20-Newsgroups dataset after applying

[1] http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-20/www/data/

[2] http://qwone.com/~jason/20Newsgroups/

the k-Means, Bisecting k-Means, and Gaussian Mixture Model algorithms, respectively. Rec.motorcycles, misc.forsale, talk.politics.guns and comp.windows.x achieve the highest F-measure values for all three clustering techniques, whereas comp.sys.ibm.pc.hardware, talk.religion.misc and soc.religion.christian had the lowest values. Moreover, Bisecting k-Means outperformed the other two clustering techniques in almost all categories.

Table IV depicts the execution times of the components (stop words removal, stemming and Tf-Idf representation) as well as the clustering algorithms applied to Webkb and 20-Newsgroups datasets. In the pre-processing step, the stop words removal was the most time-consuming process, unlike stemming, which was by far the fastest step. Furthermore, Bisecting k-Means algorithm is the fastest clustering method, while the classic k-Means is the slowest option.

Figure 2 illustrates the execution time of the three implemeted algorithms, namely, k-Means, Bisecting k-Means, and Gaussian Mixture Model for different document sizes and fixed value of k. We can observe that Bisecting k-Means is more efficient as the number of input data increases, while for smaller collections of documents, it is slower against the classic k-Means. The Gaussian Mixture Model is by far the slowest method of clustering.

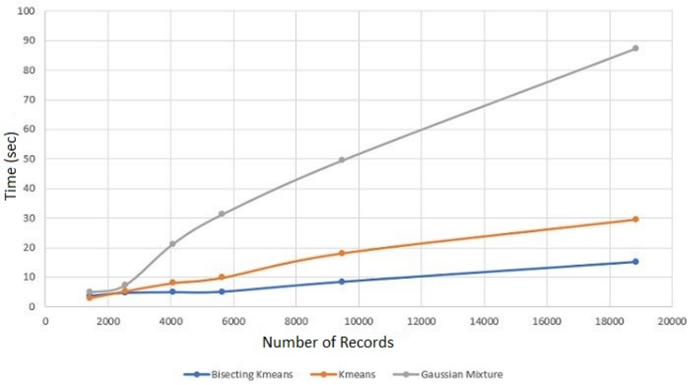

Fig. 2. Execution Time in terms of Number of Records

V. Discussion and Conclusions

This work aims to implement a method for effectively extracting textual information. This extraction has been proved in terms of efficiency. The effectiveness lies in the fact that the system was implemented in a cloud computing environment, namely Apache Spark. At the same time, its efficiency is related to the clustering methods, which were evaluated with use of F-measure. The experiments were performed on the Webkb and 20-Newsgroups datasets, which gave very satisfactory results concerning the values of F-measure exploiting the clustering algorithms of the Apache Spark Machine Learning Library. In addition, the Bisecting k-Means algorithm showed slightly better results than the traditional k-Means and the Gaussian Mixture Model algorithms.

The adoption of pre-processing techniques constitutes an essential step in enhancing the performance of clustering algorithms. In particular, removing stop words and applying stemming in each documents collection seemed to significantly improve the quality of the extracted information. At the same time, these components contributed to the temporal improvement of clustering due to input dimensionality reduction.

More to the point, the utilization of Apache Spark capabilities significantly contributed to the rapid parallel processing of large volumes of documents. That is reflected in the execution time of the algorithms. Regarding the selected clustering methods, Bisecting k-Means was the best choice in terms of F-measure and execution time perspectives. Its performance was impacted when the number of documents was small (less than 2.000 documents); in these cases, k-Means was slightly faster. The performance of Bisecting k-Means is notably improved as the collection size increases, making it suitable for large-scale data applications. Finally, Gaussian Mixture Model presented in all cases the highest execution times.

Directions for further investigation in this line of research would be the achievement of a better trade-off between response time and clustering of the documents. Moreover, more promising clustering methods will be investigated, benefiting from deep neural networks in the aforementioned techniques. Modern techniques and technologies in many important data applications (e.g. MapReduce, Dryad, MongoDB, HBase and Cassandra) can not solve the real problems of storing and exploring big data; Therefore, despite the efforts to tackle the problem of storing and processing big data in cloud environments, some important aspects of storing and processing big data have not yet been resolved and tackling it is a challenge of great interest.

Acknowledgement

This research was funded by the European Union and Greece (Partnership Agreement for the Development Framework 2014-2020) under the Regional Operational Programme Ionian Islands 2014-2020, project title: ”Indirect costs for project ”TRaditional corfU Music PresErvation through digiTal innovation””, project number: 5030952.

References

[1] M. Afzali and S. Kumar. Text document clustering: Issues and challenges. In *International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)*, pages 263–268, 2019.

[2] R. Agrawal, S. Gollapudi, A. Halverson, and S. Ieong. Diversifying search results. In *2nd International Conference on Web Search and Web Data Mining (WSDM)*, pages 5–14, 2009.

[3] A. Angel and N. Koudas. Efficient diversity-aware search. In *ACM SIGMOD International Conference on Management of Data (SIGMOD)*, pages 781–792, 2011.

[4] R. Baeza-Yates and B. A. Ribeiro-Neto. *Modern Information Retrieval: The Concepts and Technology behind Search, Second Edition*. Pearson Education Ltd., Harlow, England, 2011.

[5] A. C. Benabdellah, A. Benghabrit, and I. Bouhaddou. A survey of clustering algorithms for an industrial context. *Procedia Computer Science*, 148:291–302, 2019.

[6] E. Dritsas, I. E. Livieris, K. Giotopoulos, and L. Theodorakopoulos. An apache spark implementation for graph-based hashtag sentiment classification on twitter. In *22nd Pan-Hellenic Conference on Informatics*, pages 255–260, 2018.

TABLE III
20-Newsgroups Dataset: F-Measure Evaluation in terms of Clustering Methods

Category	k-Means	Bisecting k-Means	Gaussian Mixture Model
alt.atheism	0.71	0.79	0.73
comp.graphics	0.73	0.83	0.73
comp.os.ms-windows.misc	0.70	0.76	0.67
comp.sys.ibm.pc.hardware	0.65	0.68	0.68
comp.sys.mac.hardware	0.69	0.71	0.61
comp.windows.x	0.85	0.83	0.77
misc.forsale	0.90	0.90	0.85
rec.autos	0.77	0.84	0.81
rec.motorcycles	0.91	0.91	0.88
rec.sport.baseball	0.82	0.89	0.83
rec.sport.hockey	0.76	0.82	0.76
sci.crypt	0.70	0.76	0.74
sci.electronics	0.81	0.88	0.85
sci.med	0.74	0.83	0.76
sci.space	0.87	0.95	0.88
soc.religion.christian	0.68	0.80	0.64
talk.politics.guns	0.85	0.93	0.86
talk.politics.mideast	0.80	0.87	0.78
talk.politics.misc	0.72	0.81	0.75
talk.religion.misc	0.65	0.78	0.71

TABLE IV
Components and Algorithms Execution Time (sec)

Algorithm	Webkb	20-Newsgroups
Stop Words Removal	287.79	233.69
Stemming	5.85	4.98
Tf-Idf Representation	58.79	18.64
k-Means	64.34	29.56
Bisecting k-Means	24.27	15.08
Gaussian Mixture Model	92.54	87.83

[7] E. Dritsas, G. Vonitsanos, I. E. Livieris, A. Kanavos, A. Ilias, C. Makris, and A. K. Tsakalidis. Pre-processing framework for twitter sentiment classification. In *15th International Conference on Artificial Intelligence Applications and Innovations (AIAI)*, volume 560, pages 138–149, 2019.

[8] M. H. Dunham. *Data Mining: Introductory and Advanced Topics*. Prentice-Hall, 2002.

[9] P. Ferragina and U. Scaiella. TAGME: on-the-fly annotation of short text fragments (by wikipedia entities). In *19th Conference on Information and Knowledge Management (CIKM)*, pages 1625–1628, 2010.

[10] M. Z. Hossain, M. N. Akhtar, R. B. Ahmad, and M. Rahman. A dynamic k-means clustering for data mining. *Indonesian Journal of Electrical Engineering and Computer Science*, 13(2):521–526, 2019.

[11] R. Janani and S. Vijayarani. Text document clustering using spectral clustering algorithm with particle swarm optimization. *Expert Systems with Applications*, 134:192–200, 2019.

[12] A. Kanavos, A. Georgiou, and C. Makris. Estimating twitter influential users by using cluster-based fusion methods. *International Journal on Artificial Intelligence Tools*, 28(8):1960010:1–1960010:26, 2019.

[13] A. Kanavos, P. Kotoula, C. Makris, and L. Iliadis. Employing query disambiguation using clustering techniques. *Evolving Systems*, 11(2):305–315, 2020.

[14] A. Kanavos, C. Makris, Y. Plegas, and E. Theodoridis. Extracting knowledge from web search engine using wikipedia. In *14th International Conference on Engineering Applications of Neural Networks (EANN)*, pages 100–109, 2013.

[15] A. Kanavos, C. Makris, Y. Plegas, and E. Theodoridis. Ranking web search results exploiting wikipedia. *International Journal on Artificial Intelligence Tools (IJAIT)*, 25(3):1–26, 2016.

[16] A. Kanavos, E. Theodoridis, and A. K. Tsakalidis. Extracting knowledge from web search engine results. In *IEEE 24th International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 860–867, 2012.

[17] S. Kim and J. Gil. Research paper classification systems based on TF-IDF and LDA schemes. *Human-centric Computing and Information Sciences*, 9:30, 2019.

[18] W. Lu. Improved k-means clustering algorithm for big data mining under hadoop parallel framework. *Journal of Grid Computing*, 18(2):239–250, 2020.

[19] X. Meng, J. K. Bradley, B. Yavuz, E. R. Sparks, S. Venkataraman, D. Liu, J. Freeman, D. B. Tsai, M. Amde, S. Owen, D. Xin, R. Xin, M. J. Franklin, and R. Zadeh. Mllib: Machine learning in apache spark. *Journal of Machine Learning Research*, 17:34:1–34:7, 2016.

[20] T. K. Moon. The expectation-maximization algorithm. *IEEE Signal Processing Magazine*, 13(6):47–60, 1996.

[21] J. Mughal. Data mining: Web data mining techniques, tools and algorithms: An overview. *International Journal of Advanced Computer Science and Applications (IJACSA)*, 9(6), 2018.

[22] Z. Nasim and S. Haider. Cluster analysis of urdu tweets. *Journal of King Saud University - Computer and Information Sciences*, 3(2):2170–2179, 2020.

[23] N. Ntaliakouras, G. Vonitsanos, A. Kanavos, and E. Dritsas. IEEE an apache spark methodology for forecasting tourism demand in greece. In *10th International Conference on Information, Intelligence, Systems and Applications (IISA)*, pages 1–5, 2019.

[24] E. Patel and D. S. Kushwaha. Clustering cloud workloads: K-means vs gaussian mixture model. *Procedia Computer Science*, 171:158–167, 2020.

[25] A. S. Ramkumar and R. Nethravathy. Text document clustering using k-means algorithm. *International Research Journal of Engineering and Technology (IRJET)*, 6:1164–1168, 2019.

[26] A. Rashid and A. Chaturvedi. Cloud computing characteristics and services: A brief review. *International Journal of Computer Sciences and Engineering (IJCSE)*, 7(2):421–426, 2019.

[27] H. Raviv, O. Kurland, and D. Carmel. Document retrieval using entity-based language models. In *39th International Conference on Research and Development in Information Retrieval (SIGIR)*, pages 65–74, 2016.

[28] S. Salloum, R. Dautov, X. Chen, P. X. Peng, and J. Z. Huang. Big data analytics on apache spark. *International Journal of Data Science and Analytics*, 1(3-4):145–164, 2016.

[29] T. H. Sardar and Z. Ansari. An analysis of mapreduce efficiency in document clustering using parallel k-means algorithm. *Future Computing and Informatics Journal*, 3(2):200–209, 2018.

[30] Y. Sato, K. Izui, T. Yamada, and S. Nishiwaki. Data mining based on clustering and association rule analysis for knowledge discovery in multiobjective topology optimization. *Expert Systems with Applications*, 119:247–261, 2019.

[31] T. Tarczynski. Document clustering - concepts, metrics and algorithms. *International Journal of Electronics and Telecommunications*, 57(3):271–277, 2011.