Comparative Study of Clustering Algorithms in Text Mining Context

Abdennour Mohamed JALIL, Imad HAFIDI, Lamiae ALAMI, ENSA Khouribga

Laboratoire IPOSI

***Abstract* — The spectacular increasing of Data is due to the appearance of networks and smartphones. Amount 42% of world population using internet [1]; have created a problem related of the processing of the data exchanged, which is rising exponentially and that should be automatically treated. This paper presents a classical process of knowledge discovery databases, in order to treat textual data. This process is divided into three parts: preprocessing, processing and post-processing. In the processing step, we present a comparative study between several clustering algorithms such as KMeans, Global KMeans, Fast Global KMeans, Two Level KMeans and FWKmeans. The comparison between these algorithms is made on real textual data from the web using RSS feeds. Experimental results identified two problems: the first one quality results which remain for algorithms, which rapidly converge. The second problem is due to the execution time that needs to decrease for some algorithms.**

***Keywords* — Algorithms, Clustering, Data, Text Mining.**

I. Introduction

The advent of smartphones, social networks and cloud computing has added to the amount and sparse of data creation in the world, so much so that 90% of the world's total data has been created in the last 5 years and 70% of it by individuals. Studies predict that approximately 4 trillion gigabytes of data will exist on earth. As the world becomes increasingly digital, new techniques are requested, needed to search, analyze, and understand these huge amounts of unstructured data. This requires an automatic processing for unstructured data. This is BIGDATA R&D problematic, more specifically in the field of textual Data researches.

Text mining is a set of techniques, which aim to process those huge amounts of data and gain value from it. Introduced by Ronen and Dagan as KDT [2], we find as main branches of text mining: text extraction, summarizing, categorization, etc.

In opposite of Data mining, KDT aims to process unstructured texts, complex and over dimensioned data. Generally KDT is based on an automatic process to analyze the entire contents.

The paper is organized on three sections: In the first section, we present a text clustering system for KDT. In the second section, a number of classification algorithms are described. The third section presents a comparative study of clustering algorithms in a KDT context. At the last section some conclusions are drawn.

II. Existing Clustering System

It is generally based on automatic method to analyze and process the whole text. Among these methods there is the process show in [10], which is spread over three main stages:

A. Preprocessing

The aim of this step is to clean data and reduce noise: [11]:

1. Removal of empty words (defined articles, punctuation marks, etc).
2. Lemmatization: It is a lexical analysis of words that aims to bring together a number of words in the same family sorted by root.
3. Digital transformation: in this stage the text data is converted to digital data to classify them. There are many models of digital transformation, the most used are:
   a) **Boolean Model**: The representation of the content of a document is done by using a set of list method. It represents each word of the document by a Boolean value. It can be simple and efficient if used by specialists, but it loses its effectiveness when doing research on generalized corpus (lack of user experience)
   b) **Probabilistic Model**: the representation of the content of a document is made in the context of a probabilistic method [13]. Compared to a given query, this model gives a probability estimation of document relevance.
   c) **Vector Model**: sometimes called semantic Vector [14]**.** It is the representation of a document contents through and algebraic method that consider the semantics. Very famous method, Vector model is used to represent documents in a vector shape. The application of clustering with this model becomes easy.

B. Processing

In this step, we use one of the clustering algorithms to create the corpus [15]. Alternatively we will distribute documents on several clusters. The elements of each cluster have a common characteristic. Clustering algorithms is presented in the next section.

C. Post Processing

Some works uses the ontologies to give meaning to clusters. They use labeling **[8]** and visualization of clustering results to present hierarchical relations between the resulting clusters and evaluation. They are generally looking for the most frequent terms in each cluster to present them using semantic relations of ontologies.

III. Clustering Algorithms KMeans

Partitioning Algorithms put data in a predetermined number of clusters. The clustering approaches can be divided into two categories according to their input parameters:

1. Algorithms that take explicitly the number of input clusters **K**.
2. Algorithms that take a threshold τ as an input, Used to determine indirectly the number of clusters.

DOI: 10.9781/ijimai.2016.376

KMeans is the most known algorithm in the first category. The second category includes Two-Level-KMeans for example.

All partitioning algorithms use the following concepts:

We note:

- group of data : $X = \{X1, X2, ..., Xn\}$
- partitions : $S = \{S1, S2, ..., Sk\} k \leq n$

K is the number of desired clusters, n is the cardinality of data.

It aims to minimize the distance between points belonging to each cluster (score).

$$Args \min S \sum_{i=1}^{K} \sum_{Xj \in Si} \left\| Xj - \mu i \right\|^2$$

With μi is the center of the cluster i. Xj is an element of the set.

Alternatively, we try to find clusters with minimum of inter-cluster distance and the maximum of intra-class distance.

A. Algorithms KMeans

Based on the previous concept, five partitioning algorithms are presented in the following:

a) **K Means**

The classic and most widely used algorithm is KMeans [4]. It takes two input parameters: the number of clusters K and a set of data.

This algorithm initializes arbitrarily the center of these clusters. Before running in several iterations, KMeans affected data to the nearest cluster, and the centers are recalculated at the end of each iteration. The algorithm stops once there are no more new assignments.

b) **Global-K Means**

The majority of proposed solutions previously cannot ensure the convergence to a global optimum, Global KMeans [3] proposed a new version, which overcomes this problem.

The main idea of the algorithm is to start with a single cluster that contains all the data set.

Then each iteration creating a new cluster with the center that minimizes the squared error. The algorithm stops once reached the number of clusters specified by the user.

c) **Fast Global KMeans**

This algorithm is proposed by **Jim Z.C and all** [6], to improve the Global KMeans. The fast global k-means algorithm constitutes a straightforward method to accelerate the global k-means algorithm. Suppose we are in $k - 1$ iteration, the new center (x_n) will allocate all points x_j whose squared distance from x_n is smaller than the distance d_{k-1}^{j} from their previously closest center. d_{k-1}^{j} Is the squared distance between x_j and the closest center among the $k - 1$ cluster centers Therefore, for each such data point x_j the clustering error will decrease by $d_{k-1}^{j} - |x_n - x_j|^2$

d) **Two-Level-KMeans**

Taking a threshold τ as an input parameter, the Two-level-KMeans [7] is executed in two steps. The implementation of the classic KMeans is made first. After clusters that do not verify the threshold condition, are selected for subdivision into $(r_i \div \tau) \times n$ subclasses with (r_i) is the radius of the cluster and n is the data cardinality.

e) **FW-KMeans**

FW-KMeans applies on the vector space model (VSM). Unlike KMeans that treats all elements of the set fairly, the FWK-Means uses the notion of weight to highlight the most common elements. In addition, FW-Means introduced a constant (very low value) to avoid the noise problem.

$$F(W,Z,\Lambda) = \sum_{l=1}^{k} \sum_{k=1}^{n} \sum_{i=1}^{m} w_{l,j} \lambda \beta_{l,i} \left[d(z_{l,i}, x_{j,i}) + \sigma \right]$$

Where σ is a constant, k (≤ n) is the number of clusters; β (> 1) is an exponent; W = [$w_{l,j}$] is a k × n integer matrix; $Z = [Z_1, Z_2, ..., Z_k] \in R^{k \times m}$ are the k cluster centers; $\Lambda = \left(\Lambda_1, \Lambda_2, ..., \Lambda_K \right)$ is the set of weight vectors for all clusters in which each $\Lambda_l = (\lambda_{l,1}, \lambda_{l,2}, ..., \lambda_{l,m})$ is a vector of weights for m features of the l_{th} cluster; $d(Z_{l,i}, X_{j,i}) \geq 0$ is a distance or dissimilarity measure between the j_{th} document and the center of the l_{th} cluster on the i_{th} feature. In text clustering, they use the Euclidean distance because the frequencies of terms are numeric values.

The proposal [8] FW KMeans had the aim of reducing the noise. This reduction is made by a weighting method to decrease the influence of noise, by the concentration of comparisons on the main axes.

IV. Works and Results

At first, we recover items from RSS feed in XML format. Then, we transform the whole to flat files. After removing empty words, we proceed to lemmatization stage: for each word we look for its root using the TreeTagger Tool [9] and the last part of the preprocessing is reserved to convert results into vectors Using **TF-IDF** formula:

$$TF - IDF_{ij} = TF_{i,j} \times IDF_{ij}$$

Where $TF_{i,j} = \frac{n_{ij}}{\sum_k n_{ik}}$

And $IDF_{ij} = \log(|D|) / (|d_j : t_i \in d_j|)$

Where n_{ij} is the occurrence of the word n_i in the document j. $\sum_k n_{ik}$ Is the total number of words in the document : $|D|$: total number of documents in the corpus and $|d_j : t_i \in d_j|$: Number of documents where the term t_i appears.

An example of vector is in Fig. 1

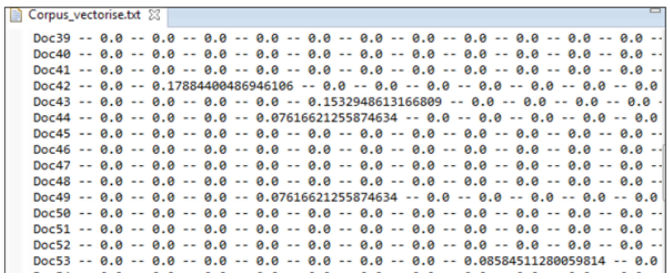

Fig 1: part of the vector file

A. Runtime Environment

The data used are taken from the actual web processed in a machine 2.5 GHz CPU (Core i5) and 8 GB of Ram, running on Windows 7.

B. Used Data

The actual used data is retrieved from several info websites (le monde, 01net, JDN etc.) Usually the documents do not have the same size (number of words), or the same natural category (political subject, computer etc.). In total 727 documents were recovered.

These documents went through a preprocessing cycle before the application of the clustering step.

C. Results

Table 1 presents obtained results using the five partitioning algorithms described previously. The time execution is also reported in this table.

TABLE 1: CLUSTERING RESULTS

Algorithm	Clusters Number	DB index	Square error	Iteration number	Execution time (s)
KMeans	10	1.469	0.867	2	0.411
	20	1.70	0.66	3	0.126
	100	2.03	0.59	2	1.247
Global KMeans	10	0.462	0.45	10	25384062
	20	0.155	0.43	20	48480265
Fast GKMeans	10	1.16	0.74	9	3344
	20	1.02	0.73	19	4454
	100	0.80	0.61	99	24627
Two-Level-KMeans	10	4.09	0.86	2	0.455
	20	1.41	0.75	1	0.517
	100	1.23	0.69	3	4.204
FW-KMeans	10	1.73	0.46	9	321400
	20	1.41	0.46	19	1995788
	100	0.75	0.42	53	29686708

All algorithms are launched five times (except Global KMeans: one time) and took the average value of the index in Tables 1. For the Two-Level- KMeans algorithm, we fix threshold as the average value of all vectors. The initialization of the centers of all algorithms (except Global KMeans) is made arbitrarily. For the β parameter of FW-KMeans, it was set at 1.5.

By analyzing the results in Table 1, we find that the classic KMeans is the fastest in terms of execution time.

This rapid convergence to a local optimum lets results become heuristic.

For the global KMeans present the longest (execution time). It is normal because it treats all possible cases. The application of this method in a real-time context is not possible. Fast Global KMeans is very quick in comparison to the Global KMeans, and gives an interesting result compared to the execution time. Two level KMeans clustering is very useful if it used for large volume of data.

A reduction in the processing time is confirmed. This reduction according to the literature [7] maintains the quality of the clustering. But it was tested on uniform data. In our case we tested them on various data. This application showed that the two level KMeans is inefficient in this case.

The FW-KMeans presents good results compared to the majority of tested algorithms. It takes into account the concept of weighting in its partitioning which increases the quality of the results but the execution time is bit little high.

V. Conclusion

In this paper, we have implemented a process of text mining. Initially, we have performed a preprocessing on the data from the web, and then we applied five clustering algorithms (KMeans Global KMeans, Fast Global KMeans, Two-level KMeans and FW-KMeans) on the data. The evaluation of the classification results is performed with Squar erreur and DBIndex.

We found similar results in the literature on our data: rapid convergence of the KMeans clustering and less performance of two level KMeans clustering without having good quality. Medium or high quality with Global KMeans, fast GKMeans and FWKmeans, but with a long execution time.

It has been found that Two-level-KMeans is ineffective in a KDT context, and its results are closed to KMeans ones. In addition, the choice of the threshold value is still a problem

The FW KMeans improves the k-means algorithm by adding a new step, in which the weights of features for different groups are calculated. The experiences show that FW KMeans can deal with a large and sparse data. The results of own experiment on text data provided from the real world show that the FW KMeans is better than the TWO Level KMeans.

The indices used for clustering evaluation are less specific and depend on the treated area. In future research, we will work on two axes:

- Integration of ontologies in different stages of the process to improve our results in terms of quality and execution time.
- The parallelization algorithms for the different stages.

References

[1] WeAre Social, http://wearesocial.fr/blog/2015/01/digital-social-mobile-les-chiffres, 2015/ 2015.

[2] Ronen and Dagan: Knowledge Discovery in Textual Databases (KDT). 2005.

[3] Aristidis Likas, Nikos Vlassis, Jakob J. Verbeek: The Global K-Means Clustering Algorithm., 2003.

[4] J. B. MacQueen: Some Methods for classification and Analysis of Multivariate Observations. 1967.

[5] Kaufman, L. And Rousseeuw, P.J. Clustering by means of Medoids, in Statistical Data Analysis Based on the L_1–Norm and Related Methods, edited by Y. Dodge, north-Holland, 1987. pp.405–416.

[6] Jim Z.C. Laia, Tsung-Jen Huanga: Fast global KMeans clustering using cluster membership and inequality. 2009.

[7] Radha Chitta, M. Narasimha Murty JournalPattern Recognition archive Volume 43 Issue 3, March 2010 pp. 796-804.

[8] Liping Jing, Michael K. Ng, Xinhua Yang, Joshua Zhexue Huang : A Text clustering System based on k-means Type Subspace Clustering and Ontology, World Academy of Science, Engineering and Technology Vol: 2 2008

[9] Helmut Schmid, Improvements in Part-of-Speech Tagging with an Application to German. 2005.

[10] Joel Azzopardi, Christopher Staff, Incremental Clustering of News Reports, 2012.

[11] IBEKWE-SANJUAN Fidelia, SANJUAN Eric, Ingénierie linguistique et Fouille de Textes, 2007.

[12] Davies, DavidL Bouldin W, A Cluster Separation Measur. IEE Transaction on Pattern Machine Intelligence. PAMI-1(2), 2012, pp.224-227

[13] [Stephen E.Roberston ; Karen Sparrck Jones], Revelence weighting of search terms, Journal of the American Society, vol 27, n°3, main-juin 1976, pp.129-146

[14] [G.Salton, A.Wong, C.S Yang], Avector space model for automatic

indexing, Communication of the ACM, v18 n°11, 1975, pp. 613-620

[15] Khan K, Sahai A. A fuzzy c-means bi-sonar-based Metaheuristic Optimization Algorithm, International Journal of Interactive Multimedia and Artificial Intelligence. 2012;1(7), pp.26-32

Mohamed Abdennour JALIL was born in Khouribga, Morocco, in 1989. He received the Bachelor degree in mathematics and computer sciences from the University of Hassan I, Morocco, in 2011, and the Master degree in computer sciences from the national school of applied sciences (ENSA), Khouribga, Morocco, in 2013. In 2015, he joined the Department of Computer sciences Engineering, University of Hassan first, as a PhD student, Abdennour worked as technical consultant in ERP industry (implementing Microsoft Dynamics Ax) at Accenture Services Morocco & TVH Consulting(2013-2015). He consulted on a variety of projects, involving quantitative analysis, Processing Huge amounts of unstructured Data and Reporting. He developed an interest in Big Data, Text Mining and sophisticated clustering algorithms.

Imad HAFIDI PhD is a professor (since 2009) and the Head of IT and Telecom Department, at the national school of applied sciences Khouribga. He gets accreditation to supervise research in 2013. Before coming to Khouribga, he completed many programs: PhD degree (2005) in Applied Mathematics at the National institute of applied sciences Lyon (INSA), and the MS degree (2008) in Software Engineering at the School of Mines Saint-Etienne. Before that he received the MS degree in numerical analysis (2001) from Lyon I university, and the bachelor degree in Applied Mathematics (1999) from the University of Hassan II Ain Chock. His research focuses on processing heterogeneous data, Text Mining and Big Data.

Lamiae ALAMI, is a PhD student working at National School of Applied Sciences Khouribga. She received the Master degree in Business intelligence and statistics from the University of Lyon II, France, in 2010.Prior to beginning the PhD program, Lamiae worked as a Business intelligence consultant in the insurance Industry at Sopra Group. She consulted on a variety of projects for France's largest insurance companies (La Mutuelle Générale, IPECA prévoyance…), involving Data warehousing, Master Data Management (MDM), building repositories, KPI's and Dashboards. Her research interests cover conception and optimization of Data warehouses, Data cleansing & integration algorithms and text Mining.