OLAP Textual Aggregation Approach using the Google Similarity Distance

Mustapha Bouakkaz

LIM Laboratory,
Laghouat University, Algeria
E-mail: m.bouakkaz@lagh-univ.dz

Sabile Loudcher

ERIC Laboratory,
Lyon 2 University, France
E-mail: sabine.loudcher@univ-lyon2.fr

Youcef Ouinten

LIM Laboratory,
Laghouat University, Algeria
E-mail: ouinteny@lagh-univ.dz

Abstract: Data warehousing and On-Line Analytical Processing (OLAP) are essential elements to decision support. In the case of textual data, decision support requires new tools, mainly textual aggregation functions, for better and faster high level analysis and decision making. Such tools will provide textual measures to users who wish to analyse documents online. In this paper, we propose a new aggregation function for textual data in an OLAP context based on the K-means method. This approach will highlight aggregates semantically richer than those provided by classical OLAP operators. The distance used in K-means is replaced by the Google similarity distance which takes into account the semantic similarity of keywords for their aggregation. The performance of our approach is analyzed and compared to other methods such as Topkeywords, TOPIC, TuBE and BienCube. The experimental study shows that our approach achieves better performances in terms of recall, precision,F-measure complexity and runtime.

Keywords: OLAP, Textual Aggregation, Google Similarity, K-means

1 Introduction

The decision process in many sectors such as health, safety, security and transport is a complex process with many uncertainties. In such cases, the decision makers require appropriate tools for diagnosis so as to perform, validate, justify, evaluate and correct the decisions they have to take. Online Analytical Processing (OLAP) has emerged to assist users in the decision making process. The model building in OLAP is based on the multidimensional structure which facilitates the visualization and the aggregation of data. This model represents both the subjects to analysis (facts), the indicators to assess the facts (measures) and the features to be analysed (dimensions). A dimension can also have a hierarchy with different levels. In order to navigate into data, there are OLAP operations such as roll-up and drill-down. With a roll-up operation a user can change the granularity of data and an aggregation function is needed to aggregate the measure. Many functions, such as maximum, minimum, average are applied to aggregate data according to the level of detail, by changing the granularity. As shown in the example of the figure 1, a decision maker analyses the number of scientific papers published by laboratories in each month. In order to have a top level view, he changes the granularity level by presenting them per each year. That means, the monthly values are aggregated into a value for each year.

According to (1), OLAP has robust solutions for numerical data. However, (2) and (3) assert that only 20% of corporate information system data are used and exploited, whereas the rest of useful information is non-additive data such as textual data. These evolutions in the characteristics and in the nature of data make OLAP tools unsuitable for most new types of data. For example textual data are out of reach of OLAP analysis. Recently, document warehousing (a set of approaches for analysis, sharing, and reusing unstructured data, such as textual data or documents) has become an important research field. Many issues are still open but we are mainly interested in taking into account the textual content of data in the OLAP analysis. In this case, adapted aggregation functions for textual measure are needed.

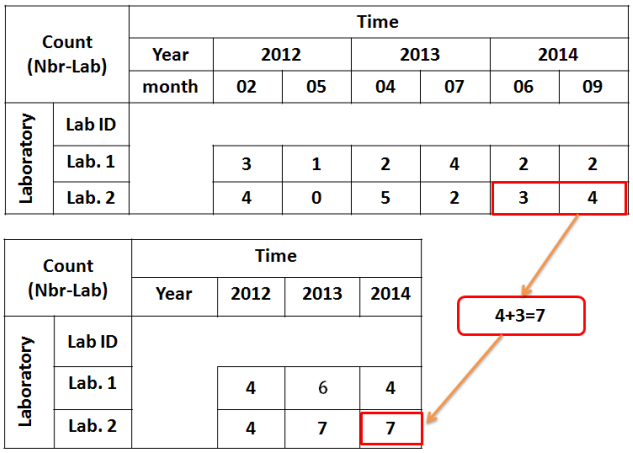

Figure 1 Multidimensional analysis of scientific papers.

Our main contribution in this paper is to provide an OLAP aggregation function for textual measures. This function allows an analysis based on keyword measures for a multidimensional document analysis. From the literature of keywords aggregation, we cluster the existing methods into four groups. The first one is based on linguistic knowledge, the second one on external knowledge, the third is based on graphs, while the last one is based on statistical methods. Our approach falls in the latter category. The existing approaches using statistical methods focus mainly on the frequencies of keywords. However, the approach that we propose uses a well known data mining technique, which is the k-means algorithm, with a distance based on the *Google similarity distance*. The *Google similarity distance* has been proposed by Google and has been tested in more than eight billion of web pages (4). The choice of this distance is motivated by the fact that it takes into account the semantic similarity of keywords.
We name our approach GOTA Google similarity distance in OLAP Textual Aggregation. The performance of our approach is designed and compared to other methods such as TopKeywords (5), BienCube (6), TuBE (7) and TOPIC(8).The last approach use the k-bisecting clustering algorithm with the Jensen-Shannon divergence for the probability distributions. The rest of the paper is organized as follows: Section 2 is devoted to related work to textual aggregation. In Section 3, we introduce our proposed approach. In Section 4, we present the experimental study which includes a comparison with some other. Finally, Section 5 concludes the paper and provides future developments.

2 Related work

In this section we provide a new classification of the existing aggregation approaches. we classify contributions found in the literature into two major categories, approaches based on the data structure such as the proprieties of the data cube and the ones based on content. Approaches that belong to the last category are classified into four sub categories, approaches based on linguistic knowledge, approaches based on external knowledge, approaches based on graph and the ones based on statistical informations.Details on this classification are developed in the following sub sections.

2.1 Approaches based on data structure

The DocCube suggested by Mothe et al.(9) is used to examine and envisage the whole document in a corpus using classification approaches. It treats several facts of document as dimensions. The major characteristic of DocCube lies in the nature of the content of the fact table. This last contains links between documents and a fact row. These links are defined by their weights according to the degree of confidence on the association (Doc, Ref). The multidimensional visualization provided in DocCube gives a user the possibility to know the relatedness between the documents and gives him a direct access to explore the document content. By exploring the dimensions, the user can view the distribution of the documents according to their URL and can manipulate the level of aggregation for his visualization. He also have direct access to the documents associated with the selected dimension values via the links provided.
Topic Cube :The analyse of text using OLAP must support the drill-down or roll-up when we want to analyse a text data on a topic dimension. Zhang et al. (10) proposed an approach called Topic Cube, the main idea of a topic cube is to use the hierarchical topic tree as the hierarchy for the text dimension. This structure allows to a user to drill-down and roll-up along this tree and discover the content of the text documents in order to view the different granularities and levels of topics in the cube. The first level in the tree contains the detail of topics, the second level is more general and last level contains the aggregation of all topics. A textual measure is needed to aggregate the textual data. The authors proposed two types of textual measures, word distribution and topic coverage. The topic coverage computes the probability that a document contains the topic. These measures allow users to know which topic is dominant in the set of documents by aggregating the coverage over the corpus. The perspectives of Zhang et al. cited in (10), are realized in a new extension called iNextCube (Information Network-Enhanced Text Cube) proposed by Yu et al. (11). They used information network analysis to automatically construct the topic hierarchy.
The Document Cube : Seng et al. (12) proposed an approach for multidimensional analysis on scientific documents. Many data extracted from the scientific articles are used as dimensional data, such as, the keywords, names of authors, title, name of conference or journal and date. However, in their works they don't explain how the keywords and the metadata

are structured in a hierarchical order. To explore the dimensional data they propose a textual measure, they associate to each document an identifier and the number of similar documents in order to facilitate rolling up and drilling down and to to ease the navigation in the different granularities and perspectives. A query result is a text cube, where cells contain the identifiers of corresponding documents stored in the corpus. A new extension for Document cube is proposed by Tseng et. al.(13) which resulted in a new query language specially designed for the document cube called MD2X (MultiDimensional Document eXpression).

Text Cube : In order to introduce the semantic aspect in the textual aggregation, Lin et al.(14) proposed an approach for data cube called text cube. The main idea is to give the user the possibility to make a semantic navigation in data dimension. To specify the semantic level in the text cube, they proposed an hierarchy where the extracted keywords represent the nodes at the base level, the ancestor nodes at upper levels are more general than children at lower level, and the nodes at top level contain terms of the corpus. The use of textual measures pull-up or push down facilitates the navigation in the hierarchy. Thus the measures, term frequency and inverted index are used for aggregated text data.

R-Cube : For Perez et al. (15)(16)(17)(18) It is a very important task to integrate structured and textual data in the same data warehouse. To get that the authors proposed architecture for a decision support system called contextualized warehouse, which allows users to obtain knowledge from all their heterogeneous data and documents and by analyse data under different contexts. Due to the variation of data, it is recommended that users specify the analytical context by providing a list of keywords, and then an R-cube (Relevance Cube) is retuned by retrieving the documents and the facts related to the selected context. In R-cube the fact is linked to the contexts, and has a dimension value correspond to the relevance with respect to the specified context. The construction of R-cube is started by evaluating the document warehouse; the result is a set of documents. Second, select the facts described by each document according to their frequency. Then, each document is assigned to those facts of the corporate data warehouse whose dimension values can be rolled-up or drilled-down. Finally, the relevance value of each fact is calculated.

2.2 Approaches based on content

The approaches, which describe Document warehousing through the most representative keywords, found in the literature can be classified into four categories. The first one is based on linguistic knowledge; the second one is based on the use of external knowledge, the third one is based on graphs, while the last uses statistical methods. The approaches based on linguistic knowledge consider a corpus as a set of the vocabulary mentioned in the documents; but the results in this case are sometimes ambiguous. To overcome this obstacle, techniques based on lexical knowledge and syntactic knowledge previews have been introduced. In (19)(20) the authors described a classification of textual documents based on scientific lexical variables of discourse. Among these lexical variables, they chose nouns because they are more likely to emphasize the scientific concepts, rather than adverbs, verbs or adjectives.

The approaches based on the use of external knowledge select certain keywords that represent a domain. These approaches often use models of knowledge such as ontology. Ravat et al. proposed an aggregation function that takes as input a set of keywords extracted from documents of a corpus and outputs another set of aggregated keywords (3). They assumed that both the ontology and the corpus of documents belong to the same domain. Oukid et al. (21) proposed an aggregation operator Orank (OLAP rank) that aggregates a set of documents by ranking them in a descending order using a vector space representation. Subhabrata et al. in (22) propose a textual aggregation model using ontology. They propose an approach to construct keywords Ontology Tree.

The approaches based on graphs, use keywords to construct graphs, where each node represents a keyword obtained after pre-processing and candidate selection. An edge represents the strength of relatedness (or semantic relatedness) between two keywords. After the graph representation step, different types of keywords-ranking approaches have been tried. The first proposed is an approach called TextRank (24), where, the edges represent the co-occurrence relations between the keywords. The idea of this approach is that, if a keyword is linked to a large number of other keywords, then it is considered as important (24). It constructs the term graph, in which the links between terms reflect their semantic relatedness, which are calculated by the term co-occurrences in the corpus. TextRank, still tends to extract high-frequency terms as keywords because these terms have more opportunities to get linked with other terms and obtain higher PageRank scores. Moreover, TextRank usually constructs a term graph using term co-occurrences as an approximation of the semantic relations between words. This will introduce much noise because of connecting semantically unrelated words and highly influence extraction performance. Other approaches have been based on TextRank in order to improve it, as ExpandRank (25) which uses a small number of neighbour documents to provide more information of term relatedness for the building on term graphs. Another potential approach to alleviate vocabulary gap is the latent topic models that learn topics from a collection of documents. The semantic relatedness between a term and a document can be estimated using the similarities of their topic distributions. The similarity scores can be used as the ranking criterion for keywords extraction (26).

TAG: Bouakkaz et al. (27) proposed a new method which performs aggregation of keywords of documents

based on the construction of a graph using the affinities between keywords, and the construction of cycles on the graph. This function produces the main aggregated keywords out of a set of terms representing a corpus. Their aggregation approach is called TAG (Textual Aggregation by Graph). It aims at extracting from a set of terms a set of the most representative keywords for the corpus of textual document using a graph. The function takes as input the set of all extracted terms from a corpus, and outputs an ordered set, containing the aggregated keywords. The process of aggregation goes through three steps: (1) Extraction of keywords with their frequencies, (2) Construction of the affinity matrix and the affinity graph, and (3) Cycle construction and aggregated keywords selection.

The approaches based on statistical methods, use the occurrence frequencies of terms and the correlation between terms. Landauer et al. (28) proposed the method LSA (Latent Semantic Analysis) in which the corpus is represented by a matrix where the rows represent the documents and the columns represent the keywords. An element of the matrix represents the number of occurrences of a word in a document. After decomposition and reduction, this method provides a set of keywords that represent the corpus. Hady et al. in (7) proposed an approach called TUBE (Text-cUBE) to discover associations among entities. The model adopts a concept similar to data cube designed for relational databases which is applied to textual data, where cells contain keywords, and an interestingness value is attached to each keyword. Bringay et al. (6) proposed two aggregation functions, the first one is based on a new adaptive measure of Tf.Idf which takes into account the hierarchies associated to the dimensions. The second one is build dynamically and is based on clustering. Wartena et al. (29) used the k-bisecting clustering algorithm based on the Jensen-Shannon divergence of probability distributions described in (30). Their method starts by selecting two elements that are far apart as the seeds of the two first clusters. Each one of the other elements is then assigned to the cluster of the closest seed. Once all the elements have been assigned to clusters, the centres of both clusters are computed. The new centres are used as new seeds for finding new two clusters and the process is repeated until each of the two new centres converge up to some predefined precision. If the diameter of a cluster is larger than a specified threshold value, the whole procedure is applied recursively to that cluster. In (5) the authors proposed a second aggregation function called TOP-Keywords to aggregate keywords. They computed the frequencies of terms using the $Tf.Idf$ function, and then they selected the first k most frequent terms. The authors of (31) proposed the C-Value algorithm, which creates a ranking for potential keywords by using the length of the phrases which contain keywords, and their frequencies. In (32) the authors proposed a technique for extracting summary sentences for a set of documents using the weight of the sentences and the documents.

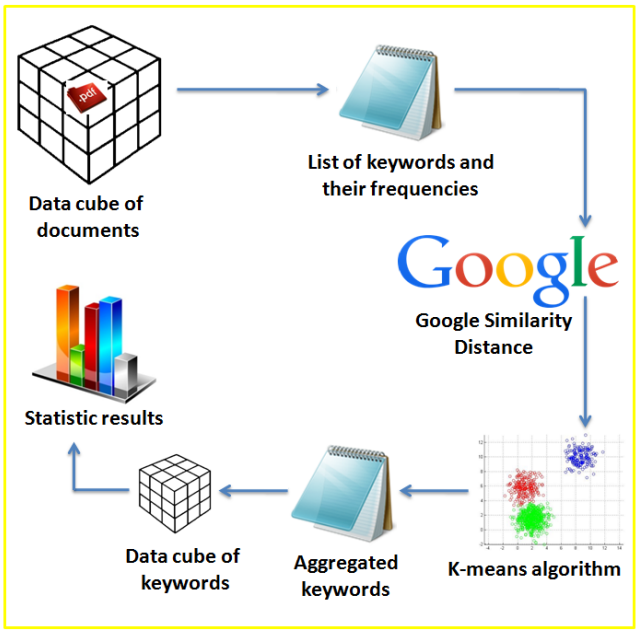

Figure 2 System architecture.

3 Proposed method

Our aim is to create a suitable environment for the online analysis of documents by taking into account textual data. In Text OLAP, the measure can be textual such as a list of keywords. When a user wants to obtain a more aggregate view of data, he does a roll-up operation which needs an adapted aggregation function. Our approach is composed of three main parts, including: (1) extraction of keywords with their frequencies; (2) construction of the distance matrix between words using the *Google similarity distance*; (3) applying the k-means algorithm to distribute keywords according to their distances, and finally (4) selection the k aggregated keywords. Figure 2 illustrates our system architecture.

3.1 Extraction of keywords

Given a corpus, the set of terms T is obtained after cleaning stop words, the lemmatization and the selection of the most significant terms. There are different ways to select such terms, we use the weight of the term because it represents the degree of its importance in the document. In our case we take the same threshold to extract pertinent terms. These weights are defined as follows:

$$\forall t_i \in T, w_i = \frac{tf_i}{\sum tf_i} \quad (1)$$

Where w_i is the weight of term t_i, tf_i is the frequency of occurrence of term t_i in the corpus.

3.2 Construction of the Google Distance Matrix

With a collection of many documents, their corresponding vectors can be stacked into a matrix.

By convention, document vectors form the rows, while the vector elements (called keywords) form the matrix columns. With n documents and m keywords, we have an nxm matrix and we will use the notation $DTM[n,m]$. An element of the matrix represents the frequency of a term j in a document i. Let $DTM(i,j) = tf_{ij}$ where tf_{ij} is the frequency of occurrence of term j in document i. We use the *Google Similarity Distance* (GSD) proposed by (4) to construct the distance matrix (GDM) between keywords. It is a symmetric square matrix where rows and columns represent the keywords. The *Google Similarity Distance*, $GSD(x,y)$ is defined as follows:

$$\frac{Max(logH(x), logH(y)) - logH(x,y)}{logN - min(logH(x), logH(y))} \quad (2)$$

The attributes $H(x)$ and $H(y)$ represent the number of term frequency of the keywords x and y, respectively. The attribute $H(x,y)$ represents the number of documents containing both x and y and N is the number of documents in the corpus.

3.3 Clustering

We use the k-means algorithm for clustering keywords into clusters. The number of clusters k is defined by the user, and it represents the number of aggregated keywords. The first step is to define k centroids, one for each cluster, by choosing K keywords that are as far apart as possible. The next step is to take each point belonging to the given data set and to associate it to the nearest centroid according to their distance in the Google distance matrix. When no point is pending, the first step is completed and we re-calculate the k new centroids of the clusters. The process is then repeated with the K new centroids. The K centroids change their location step by step until no more changes are done. The process ends up with the K clusters.
We chose the k-means method for the following reasons: (1) its result format which is a partition that corresponds to the building process of the aggregation level, and (2) its low and linear algorithmic complexity which is crucial in the context of OLAP to provide the user with quick results.

3.4 Aggregated keyword selection

After the clustering, we select from each cluster the keyword that has the highest value of H as an aggregated keyword. H is defined in the *Google Similarity Distance* (GSD)and represents the number of documents containing the keyword. Figure 3 describes the different steps of our algorithm.

3.5 Example

In our running example of scientific articles, the measure is a list of keywords. There are thirteen (13) documents $D_1, ..., D_{13}$ and ten (10) terms: {XML, OLAP, Datamining, Query, Datawarehouse, Document, System, Cube, Function, Network}. The frequency matrix is defined in Table 1. The *Google Similarity Distance* between keywords is given in Table 2. The use of k-means clustering produces the following results: C1{M2, M5}, C2{M4, M8, M10}, C3{M1, M3, M6, M7, M9}.

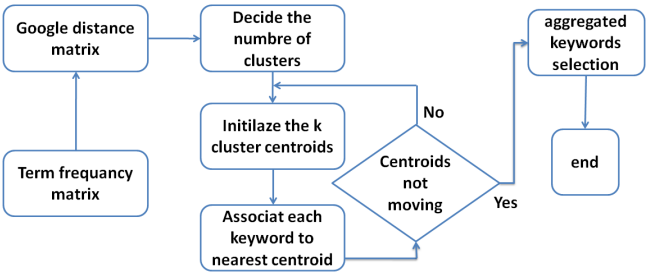

Figure 3 Steps of GOTA run.

Table 1 Document Term Matrix

	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10
D1	10	9	22	15	9	20	15	9	28	39
D2	15	22	26	0	9	16	11	0	25	0
D3	5	15	0	15	22	0	15	0	0	0
D4	0	16	0	0	15	10	0	0	0	0
D5	16	12	2	13	16	12	0	12	2	0
D6	21	0	19	21	17	9	0	0	10	0
D7	13	0	14	0	0	15	1	0	17	0
D8	17	0	8	0	0	8	0	18	20	0
D9	22	14	0	0	14	21	0	17	0	0
D10	0	7	0	0	7	0	15	18	20	0
D11	5	18	10	5	15	15	15	18	20	0
D12	20	4	7	17	4	7	0	5	3	105
D13	1	10	11	1	10	17	0	16	10	0

Table 2 Google Similarity Distance Matrix

	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10
M1	0									
M2	1.2	0								
M3	0.5	1.6	0							
M4	0.7	0.8	0.7	0						
M5	1.2	0.0	1.4	0.8	0					
M6	0.0	1.2	0.5	1.0	1.2	0				
M7	0.8	1.4	0.8	0.9	1.0	1.1	0			
M8	1.0	0.6	0.9	0.5	0.6	0.6	1.3	0		
M9	0.4	1.4	0.3	0.8	1.4	0.4	1.0	0.8	0	
M10	0.9	0.9	0.8	0.7	0.9	0.9	0.5	0.7	0.9	0

After that, we select one keyword from each cluster that has the highest value of H. If two or more keywords belonging to the same cluster have the same value of H, then we take one of them that has the highest $tf * idf$ score. The thirteen documents of the example are thus represented by the following keywords: {M5=Data Warehouse, M6=Document M8=Cube}.

4 Experimental study

4.1 Textual Benchmark

There are several available benchmarks for evaluating aggregated keywords approaches. Authors in (33) used a dataset to test their approach containing 800 journal article abstracts from Inspec[1], published between 1998 and 2002. In (34) the authors compiled a dataset containing 120 computer science articles from 4 to 12 pages. in (35) the authors developed a dataset of 308 documents taken from DUC 2001. Authors in (36) compiled a collection of 500 medical articles from PubMed[2]. In (37) the authors used 680 articles from the same source for years 2003 to 2005, with author assigned keywords. The authors in (38) collected a dataset of 100 articles from the ACM Digital Library (conference and workshop papers), ranging from 6 to 8 pages, including tables and figures. In (39) the authors proposed a tool that generates automatically a dataset using keywords assigned by users of the collaborative citation platform CiteULike[3]. These corpuses are summarized in Table 3.

Table 3 Existing benchmarks

References	Corpus size
A.Hulth (2003)	800
T.Nguyen (2007)	120
X.Wan (2008)	308
A.Schutz (2013)	500
M.Krapivin (2009)	680
K.SuNam (2013)	100

In this work we compiled two corpora of our own which are much larger than the ones mentioned above. The first corpus is from the *IIT* conference[4] (conference and workshop papers) for the years 2008 to 2014. It consists of 700 papers ranging from 7 to 8 pages in IEEE format, including tables and figures. The second corpus called Ohsumed collection [5] which includes medical reports from the MeSH categories and It consists of 20,000 documents.
The keywords are extracted from the full words using Microsoft Academic Search[6] keywords.

The keywords extraction function is based on the Microsoft Academic Search web site (*MAS*). MAS classifies scientific articles according to fifteen scientific fields by extracting the scientific keywords from articles and ordering them according to their frequencies. We use the lists of keywords produced by MAS and we choose 2000 most frequent keywords form each field as shown in Figure 4 .

The extraction of keywords from our two corpora is performed according to these chosen lists. At the end we keep only the keywords with a $tf * idf$ higher then 30%. The output of this process is the two fold matrix of

Figure 4 Steps of keywords' extraction

Documents x *Keywords*, which is used to compare our approach and the other textual aggregation approaches. For the evaluation task of the keywords aggregation, many type of measures have been proposed in (40; 41; 42). But the most used are the recall, the precision, and the F-measure. The recall is the ratio of the number of documents to the total number of retrieved documents.

$$Recall = \frac{\mid \{RelevantDoc\} \cap \{RetrievedDoc\} \mid}{\mid \{RelevantDoc\} \mid} \quad (3)$$

The precision is the ratio of the number of relevant documents to the total number of retrieved documents.

$$Precision = \frac{\mid \{RelevantDoc\} \cap \{RetrievedDoc\} \mid}{\mid \{RetrievedDoc\} \mid} \quad (4)$$

The F-measure or balanced F-score, which combines precision and recall, is the harmonic mean of precision and recall.

4.2 Results

In this section, we report an empirical study to evaluate our aggregated keyword function using two real corpora. We also compare its performance with those of (8) (7) (6) (5) .

The experimentation has been performed on a PC running the Microsoft Windows 7 Edition operating system, with a 2.62 GHz Pentium Dual-core CPU, 1.0 GB main memory, and a 300 GB hard disk. To test and compare the different approaches we have compiled two real corpora as mentioned in Section 4.1, with 600 articles, 800000 words and 2182 keywords extracted for the first corpus and 20000 articles, 1300000 words and 985 keywords extracted for the second corpus.
To perform this comparison, we use four evaluation metrics : recall, precision, F-measure and the run time for different values of k . We also give a comparison of the complexity for the five algorithms. The results are summarized in Figures 5 and 6.

Overall, our approach produces highest values of the recall, the precision and F-measure. For instance, in the case of k=3, we obtained a recall of 96% compared with 63%, 25%, 40% and 10% obtained by Topkeyword, TOPIC, BienCube and TuBE respectively. We also obtained a precession of 66% compared with 21%, 32%, 10% and 3% obtained by Topkeyword, TOPIC, BienCube and TuBE respectively. As for the F-measure, we obtained a value of 78% compared with 31%, 28%, 16% and 5% obtained by Topkeyword, TOPIC, BienCube and TuBE respectively. In the case

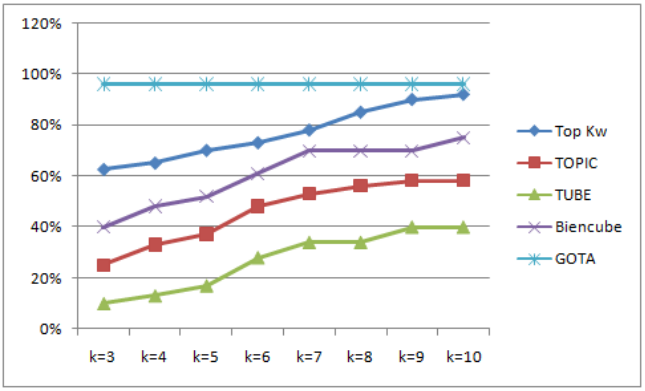

Figure 5 Comparaison between the recall - First corpus

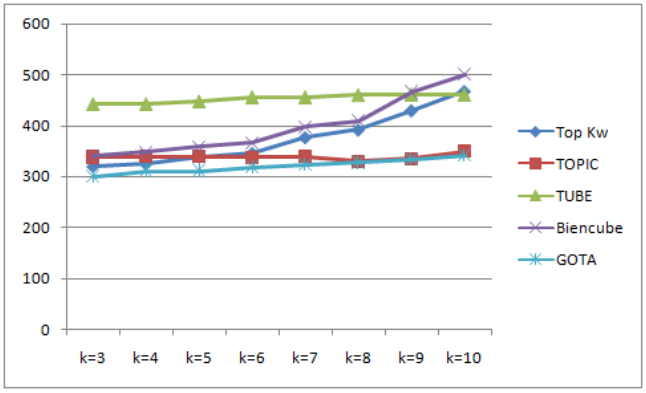

Figure 8 Comparaison between the Runtime - First corpus

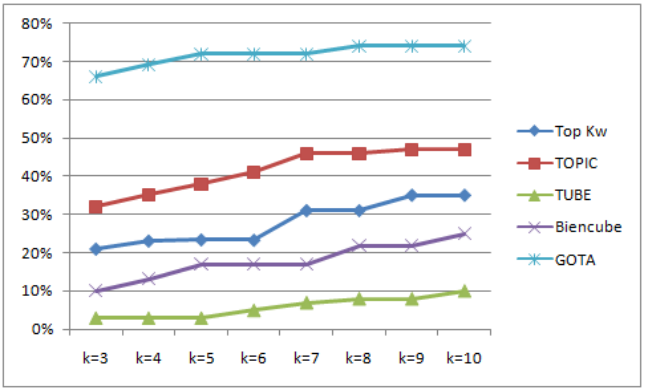

Figure 6 Comparaison between the Precession - First corpus

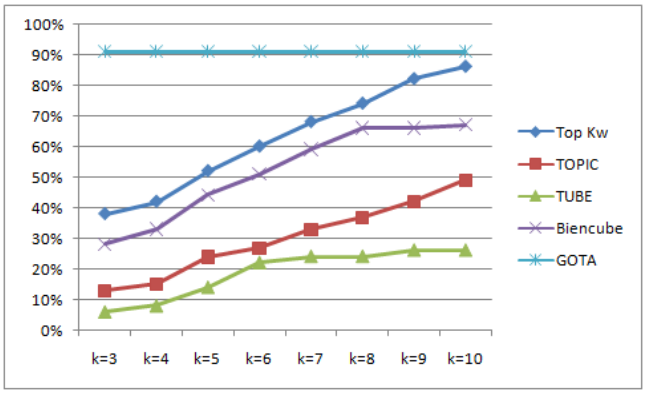

Figure 9 Comparaison between the recall - Second corpus

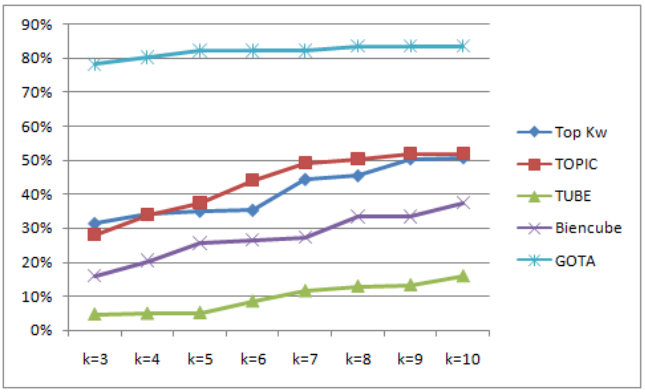

Figure 7 Comparaison between the F-mesure - First corpus

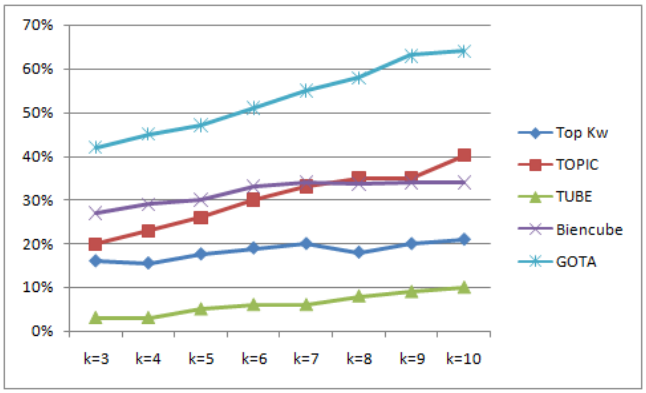

Figure 10 Comparaison between the Precession - Second corpus

of k=10, the value we obtained a recall of 96% is to be compared with 92%, 58%, 75% and 40% obtained by Topkeyword, TOPIC, BienCube and TuBE respectively. The precession obtained of 74% is to be compared with 35%, 47%, 25% and 10% obtained by Topkeyword, TOPIC, BienCube and TuBE respectively. As for the F-measure, the value of 84% is compared with 51%, 52%, 38% and 16% obtained by Topkeyword, TOPIC, BienCube and TuBE respectively. In order to determine the runtime for each approach, we carried out 10 executions of each approach.

The results obtained from the second test using a larger corpus confirm the results obtained in the first test. we note that our approach achieves better performance compared to the other approaches
The difference between the five approaches is highly noticeable in (Figure 5 and 6). This is due to the difference in the complexities of the five approaches.

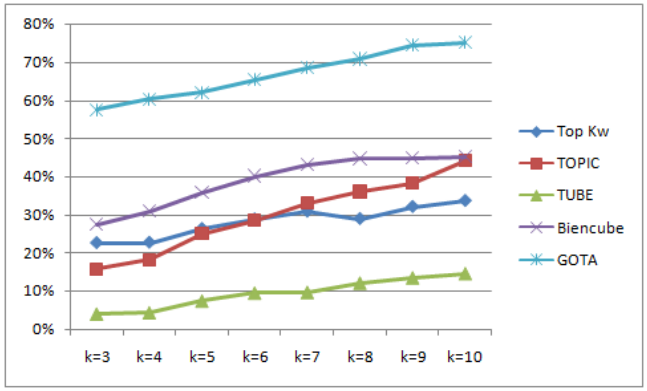

Figure 11 Comparaison between the F-mesure - Second corpus

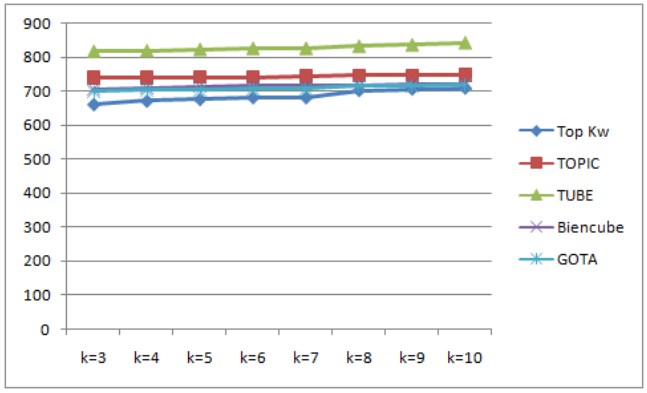

Figure 12 Comparaison between the Runtime - Second corpus

Our approach GOTA is based on k-means which has a complexity of $O(N)$. the same thing with Topkeyword and BienCube which have a complexity of $O(N)$ (5) (6). On the other hand TOPIC is based on the k-bisecting clustering which has a complexity of $O((k-1)kN)$. where k is the number of clusters and N the number of terms (8). for TUBE the complexity is $O(N^2)$ (7).

5 Conclusions

We have presented in this paper, an OLAP aggregation function for textual data. which aggregates keywords using the k-means algorithm with the Google Similarity Distance to measure semantic distances between keywords. The proposed approach GOTA was then compared with those of (5), (6), (7) and (8). The obtained results show that, overall, our approach achieves better performances in terms of recall, precision, F-measure and runtime. This work opens several promising issues and presents new challenges in the domain of aggregation in Text OLAP. Our aim in the future works is to explore the use of frequent patterns mining methods for the aggregation of textual data in an OLAP context and compare its performances with other approaches using big data sets.

References

[1] Sullivan, D.: Document Warehousing and Text Mining. John Wiley and Sons. (2001)

[2] Tseng, F. and A. Chou : The concept of document warehousing for multi-dimensional modeling of textual-based business intelligence. journal of Decision Support Systems. 42, 727–744 (2006)

[3] Ravat, F. and O. Teste and R. Tournier : OLAP Aggregation Function for Textual Data Warehouse. In International Conference on Enterprise Information Systems. 151–156 (2007)

[4] Cilibrasi, R. and P. Vitanyi: The google similarity distance. IEEE Transactions on Knowledge and Data Engineering. 370–383(2007)

[5] Ravat, F., Teste, O., Tournier, R., Zurfluh, G. (2008). Top Keyword: an aggregation function for textual document OLAP. In Data Warehousing and Knowledge Discovery (pp. 55-64). Springer Berlin Heidelberg.

[6] Bringay, S., Bchet, N., Bouillot, F., Poncelet, P., Roche, M., Teisseire, M. (2011, January). Towards an on-line analysis of tweets processing. In Database and Expert Systems Applications (pp. 154-161). Springer Berlin Heidelberg.

[7] Lauw, H. W., Lim, E. P., Pang, H. (2007, March). TUBE (Text-cUBE) for discovering documentary evidence of associations among entities. In Proceedings of the 2007 ACM symposium on Applied computing (pp. 824-828). ACM.

[8] Wartena, C. and R. Brussee : Topic detection by clustering keywords. International Conference on Database and Expert Systems Applications. 54–58 (2008)

[9] Mothe, J., Chrisment, C., Dousset, B., Alaux, J. (2003). DocCube: Multidimensional visualisation and exploration of large document sets. Journal of the American Society for Information Science and Technology, 54(7), 650-659.

[10] Zhang, D., Zhai, C., Han, J. (2009, April). Topic Cube: Topic Modeling for OLAP on Multidimensional Text Databases. In SDM (Vol. 9, pp. 1124-1135).

[11] Yu, Y., Lin, C. X., Sun, Y., Chen, C., Han, J., Liao, B., Zhao, B. (2009). iNextCube: Information network-enhanced text cube. Proceedings of the VLDB Endowment, 2(2), 1622-1625.

[12] Tseng, F. S., Chou, A. Y. (2006). The concept of document warehousing for multi-dimensional modeling of textual-based business intelligence. Decision Support Systems, 42(2), 727-744.

[13] Tseng, F. S., Lin, W. P. (2006). D-Tree: a multi-dimensional indexing structure for constructing document warehouses. Journal of Information Science and Engineering, 22(4), 819-842.

[14] Lin, C. X., Ding, B., Han, J., Zhu, F., Zhao, B. (2008, December). Text cube: Computing ir measures for multidimensional text database analysis. In Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on (pp. 905-910). IEEE.

[15] Prez, J. M., Berlanga, R., Aramburu, M. J., Pedersen, T. B. (2007, April). R-cubes: OLAP cubes contextualized with documents. In Data Engineering, 2007. ICDE 2007. IEEE 23rd International Conference on (pp. 1477-1478). IEEE.

[16] Prez, J. M., Berlanga, R., Aramburu, M. J., Pedersen, T. B. (2008). Integrating data warehouses with web data: A survey. Knowledge and Data Engineering, IEEE Transactions on, 20(7), 940-955.

[17] Prez, J. M., Berlanga, R., Aramburu, M. J., Pedersen, T. B. (2008, October). Towards a data warehouse contextualized with web opinions. In e-Business Engineering, 2008. ICEBE'08. IEEE International Conference on (pp. 697-702). IEEE.

[18] Prez-Martnez, J. M., Berlanga-Llavori, R., Aramburu-Cabo, M. J., Pedersen, T. B. (2008). Contextualizing data warehouses with documents. Decision Support Systems, 45(1), 77-94.

[19] Poudat, C., Cleuziou, G., Clavier, V. (2006). Catgorisation de textes en domaines et genres. Document numrique, 9(1), 61-76.

[20] Kohomban, U. S., Lee, W. S. (2007, January). Optimizing classifier performance in word sense disambiguation by redefining word sense classes. In Proceedings of the International Joint Conference on Artificial Intelligence (pp. 1635-1640).

[21] Oukid, L., Asfari, O., Bentayeb, F., Benblidia, N., Boussaid, O. (2013, October). CXT-cube: contextual text cube model and aggregation operator for text OLAP. In Proceedings of the sixteenth international workshop on Data warehousing and OLAP (pp. 27-32). ACM.

[22] Mukherjee, S., Joshi, S. (2014). Author-Specific Sentiment Aggregation for Polarity Prediction of Reviews. In Ninth International Conference on Language Resources and Evaluation (pp. 3092-3099). ELRA.

[24] Mihalcea, R., Tarau, P. (2004, July). TextRank: Bringing order into texts. Association for Computational Linguistics.

[24] Mihalcea, R., Tarau, P. (2004, July). TextRank: Bringing order into texts. Association for Computational Linguistics.

[25] Wan, X., Xiao, J. (2008, July). Single Document Keyphrase Extraction Using Neighborhood Knowledge. In AAAI (Vol. 8, pp. 855-860).

[26] Blei, D. M., Lafferty, J. D. (2006, June). Dynamic topic models. In Proceedings of the 23rd international conference on Machine learning (pp. 113-120). ACM.

[27] Bouakkaz, M., Loudcher, S., Ouinten, Y. (2014, November). Automatic textual aggregation approach of scientific articles in OLAP context. In Innovations in Information Technology (INNOVATIONS), 2014 10th International Conference on (pp. 30-35). IEEE.

[28] Landauer, T. K., Foltz, P. W., Laham, D. (1998). An introduction to latent semantic analysis. Discourse processes, 25(2-3), 259-284.

[29] Wartena, C., Brussee, R. (2008, September). Topic detection by clustering keywords. In Database and Expert Systems Application, 2008. DEXA'08. 19th International Workshop on (pp. 54-58). IEEE.

[30] Fuglede, B., Topsoe, F. (2004, June). Jensen-Shannon divergence and Hilbert space embedding. In IEEE International Symposium on Information Theory (pp. 31-31).

[31] Frantzi, K., Ananiadou, S., Mima, H. (2000). Automatic recognition of multi-word terms:. the c-value/nc-value method. International Journal on Digital Libraries, 3(2), 115-130.

[32] El-Ghannam, F., El-Shishtawy, T. (2014). Multi-Topic Multi-Document Summarizer. arXiv preprint arXiv:1401.0640.

[33] Hulth, A. : Improved automatic keyword extraction given more linguistic knowledge. Empirical Methods in Natural Language Processing. 216–223 (2003)

[34] Nguyen, T. and M. Kan :Key phrase Extraction in Scientific Publications. International Conference on Asian Digital Libraries. 317-326 (2007)

[35] Wan, X. and J. Xiao: CollabRank: Towards a Collaborative Approach to Single Document Keyphrase. International Conference on Computational Linguistics. 317-326 (2008)

[36] Schutz, A.: Keyphrase Extraction from Single Documents in the Open Domain Exploiting Linguistic and Statistical Methods. National University of Ireland. Master thesis (2013)

[37] Krapivin, M. and M. Marchese:Large Dataset for Keyphrases Extraction. University of Trento. Technical Report (2009)

[38] SuNam, K. and O. Medelyan and Min-Yen Kan: Automatic Keyphrase Extraction from Scientific Articles. In Language Resources and Evaluation. 723–742 (2013)

[39] Medelyan and Frank and Witten: Human-competitive tagging using automatic keyphrase extraction. Empirical Methods in Natural Language Processing. 1318-1327 (2009)

[40] Tague Sutcliffe : Measuring the informativeness of a retrieval process. Proc. of SIGIR 23–36(1992)

[41] Jones, K. and P. Willett: Readings in Information Retrieval. Morgan Kaufmann Publishing (1997)

[42] Trec: Common Evaluation Measures. The Twenty-Second Text REtrieval Conference. http://trec.nist.gov/pubs/trec22/trec2013.html (2015)