A New Tool for Textual Aggregation In Information Retrieval

Mustapha Bouakkaz[1], Sabine Loudcher[2] and Youcef Ouinten[1]
[1] *LIM Laboratory, University of Laghouat, Laghouat, Algeria*
[2] *ERIC Laboratory, University of Lyon2, Lyon, France*

Keywords: Aggregation, OLAP , Textual data, Algorithm.

Abstract: We present in this paper a system for textual aggregation from scientific documents in the online analytical processing (OLAP) context. The system extracts keywords automatically from a set of documents according to the lists compiled in the Microsoft Academia Search web site. It gives the user the possibility to choose their methods of aggregation among the implemented ones. That is TOP-Keywords, TOPIC, TUBE, TAG, BienCube and GOTA. The performance of the chosen methods, in terms of recall, precision, F-measure and runtime, is investigated with two real corpora ITINNOVATION and OHSUMED with 600 and 13,000 scientific articles respectively, other corpora can be integrated to the system by users.

1 INTRODUCTION

The huge increasing amount of complex data such as text available in different web sites, e-mails, local networks in business company, electronic news and elsewhere is overwhelming. This uncontrolled increase of information in the different fields, makes difficult to exploit the useful ones from the rest of data. This situation starts switching the information from useful to troublesome. The capability of OLAP tools available especially the text OLAP is not growing in the same way and the same speed the amount of textual documents is increasing. This problem is dramatically exacerbated by the big quantity of textual documents indexed by Search engines every moment. This makes the task of text OLAP and knowledge extraction from textual documents very limited and reduces the competitive advantage we can gain. Recently, a large number of systems have been developed over the years to solve this kind of problems and perform tasks in Information Retrieval; many of these systems perform specific tasks such as word counter and text summarization, however they are not in the level to satisfy the growing need of users to extract the useful information from documents using Text OLAP approaches.

In this paper we describe a software platform for keywords extraction and aggregation in an OLAP context. The platform implements a new way for extracting keywords from a corpus of document based on the Microsoft academia research web site and six algorithms for keyword aggregation which process a corpus of textual data to discover aggregated keywords.

The rest of the paper is organized as follows: Section 2 introduces related works in keywords extraction and aggregation in OLAP context. Section 3 describes the main components of the software prototype along with their functionalities. Whereas section 4 is devoted to numerical experiments. Finally, Section 5 presents conclusions and discusses further developments.

2 EXISTING APPROACHES AND TOOLS

Many approaches are proposed for keyword extraction but only a few for aggregation keywords. On the other hand, the majority of the existing work is based on information retrieval, and only some of them are in the OLAP context, where textual documents are stored in a data warehouse. In this section we make an inventory of the existing approaches in OLAP context, which describes a corpus of documents through the most representative aggregated keywords. There is a classical classification that includes the supervised and unsupervised approaches for keywords extraction, meanwhile in our case we introduce a new classification for textual extraction and aggregation approaches proposed in the OLAP context. We classify the previous works found in the literature into four categories. The first one uses statistical meth-

Bouakkaz, M., Loudcher, S. and Ouinten, Y.
A New Tool for Textual Aggregation In Information Retrieval.
In *Proceedings of the 18th International Conference on Enterprise Information Systems (ICEIS 2016) - Volume 1*, pages 232-237
ISBN: 978-989-758-187-8

ods; the second one is based on linguistic knowledge; the third one is based on graphs; while the last uses external knowledge.

The approaches based on statistical methods use the occurrence frequencies of terms and the correlation between terms to extract the keywords. Hady *et al.* (Hady et al., 2007) proposed an approach called TUBE (Text-cUBE). They adopted a relational database to textual data based on the cube design, each cell contains keywords, and they attached to each keyword an interestingness value. Zhang *et al.* (Zhang et al., 2009) 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 users to drill-down and roll-up along this tree. users discover also 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 contains more general types and the 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 user to know which topic is dominant in the set of documents by aggregating the coverage over the corpus. Ravat *et al.* (Ravat et al., 2008) proposed an aggregation function called TOP-Keywords to aggregate keywords extracted from documents. They used the $tf.idf$ measure, then they selected the first k most frequent terms. Bringay *et al.* in (Bringay et al., 2011) proposed an aggregation function, based on a new adaptive measure of $tf.idf$. It takes into account the hierarchies associated to the dimensions. Wartena *et al.* (Wartena and Brussee, 2008) proposed another method we called TOPIC in which they used the k-bisecting clustering algorithm and based on the Jensen-Shannon divergence for the probability distributions as described in (Archetti and Campanelli, 2006). Their method starts with the selection of two elements for the two first clusters. are assigned to the cluster of the closest of the two selected elements. Once all the terms are assigned, the process will be repeated for each cluster with a diameter larger than a specified threshold value. Bouakkz et al. (Bouakkaz et al., 2015) proposed a textual aggregation based on 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. their approach entitled GOTA 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.

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. However, to overcome this obstacle, techniques based on lexical knowledge and syntactic knowledge previews have been introduced. In (Poudat et al., 2006; Kohomban and Lee, 2007) 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 other aggregation function that takes as input a set of keywords extracted from documents of a corpus and that outputs another set of aggregated keywords (Ravat et al., 2007). They assumed that both the ontology and the corpus of documents belong to the same domain. Oukid *et al.* proposed an aggregation operator Orank (OLAP rank) that aggregated a set of documents by ranking them in a descending order using a vector space representation (Oukid et al., 2013).

The approaches based on graphs use keywords to construct a keyword graph. The nodes represent the keywords obtained after pre-processing, candidate selection and edge representation. After the graph representation step, different types of keyword ranking approaches have been applied. The first approach proposed in (Mihalcea and Tarau, 2004) is called TextRank, where graph nodes are the keywords and edges represent the co-occurrence relations between keywords. The idea is that, if a keyword gets link to a large number of other keywords, this keyword will be considered as important. Bouakkaz et al. (Bouakkaz et al., 2014) propose a new method which performs aggregation of keywords of documents based on the graph theory. 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 software system developed in this domain consists of two main components; Text Pre-processor and Topics Extractor. Text pre-processor, offers learning and inference functionalities. The learning functionality pre-processes a document collection by exploiting a stop words list and a general purpose to obtain the word-document matrix according to the bag-of-words model. The user can choose the number of words to be used for document indexing. The inference functionality processes a document to obtain one of the following bag-of-words representations; binary, term frequencies and the inverse term document frequency. Topic extractor implements a customized version of the Latent Dirichlet Allocation (LDA) model (Blei and Andrew, 2003). The solution of the LDA learning is obtained by using the Expected Maximization and the Gibbs Sampling algorithms which have been implemented in the C++ programming language on a single processor machine. Each topic is summarized through the estimate of its prior probability, a sorted list of its most frequent words together with the estimate of their conditional probabilities. Semantria [1] is a text analytical tool that offers an API that performs sentiment analysis and analytic text. Users can be integrated in the service to quickly yield actionable data from their unstructured text data, from review sites, blogs, or other sources. Additionally, users can download trial version and use Semantria for Excel, which installs directly into Office Excel to set up an environment for analyses.

3 THE SOFTWARE SYSTEM DESCRIPTION

In order to create a suitable environment for the online analysis of textual data, we intend to propose a new software which performs aggregation of keywords. The system described in this paper consists of three main components; namely Text Pre-processor, Keywords Extractor and Keywords Aggregator. These components have been integrated into a software system developed with Java programming language.

3.1 Text Pre-processor

This software component implements functionalities devoted to document pre-processing and document corpus representation. It offers words counter, and represents the documents of the corpus as a list of words with their frequencies (Figure1). Furthermore, binary and term frequency representations are allowed. The system takes the pdf, Microsoft Word and txt formats as valid inputs as shown in figure 1.

3.2 Keywords Extractor

This component is for keywords extraction. The keyword extraction function is based on the Microsoft Academic Search web site (MAS). MAS is a service provided by Microsoft to the public and it is free of charge. MAS classifies scientific articles into fifteen categories according to their fields. In each category it extracts the scientific keywords from articles and reorders them according to their frequencies. Our keywords extractor component uses this list of keywords and takes form each field the 2000 most frequent keywords, which are saved in separate text files. After that, Keywords Extractor process starts to compare MAS keywords with whole words extracted by the Text Pre-processor component. When a MAS keyword exists in the list, the extractor component saves it in a text file with its frequency and the name of the document in which it occurs.

Once our process is finished, we will get the right useful keywords validated by MAS. The output of this component is a two fold Matrix of document and keywords (MDKW). which is used by the third component to aggregate keywords.

3.3 Keywords Aggregation

The keywords aggregation component uses a set of textual aggregation algorithms TOP-Keywords, TOPIC, TUBE, TAG, BienCube and GOTA to aggregate keywords obtained in the previous step. it also produces the recall, precision, F-measure and the run time for each algorithm.

3.4 Graphical User Interface

The graphical user interface (GUI) is a necessary element in our system (OLAP-TAS) we take into consideration the ergonomical aspect to add an interactivity between the user and the machine when using our platform. The aim of the graphical user interface is to give the user a simple access to OLAP-TAS algorithms by a number of windows that help him to navigate in the system and test the different implemented algorithms without any need of previous Java programming experience or knowledge.

It is also helpful to assist students and researchers

[1] https://semantria.com/

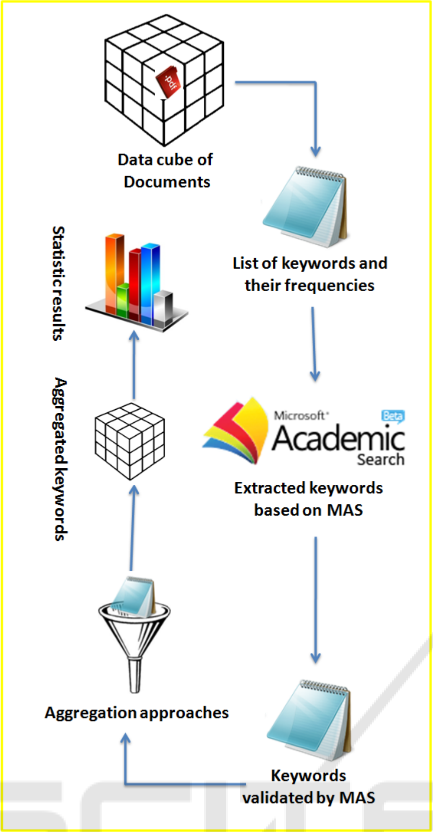
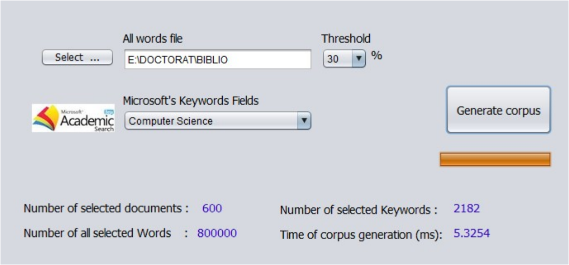

Data cube of Documents
List of keywords and their frequencies
Microsoft Academic Search Beta
Extracted keywords based on MAS
Keywords validated by MAS
Aggregation approaches
Aggregated keywords
Statistic results

Figure 1: System architecture.

to do their scientific works and research experiments in a visual platform. It is obvious that the use of an interactive tool facilitates understanding and makes learning more beneficial task for many learners.



All words file: E:\DOCTORAT\BIBLIO

Threshold: 30 %

Microsoft's Keywords Fields: Computer Science

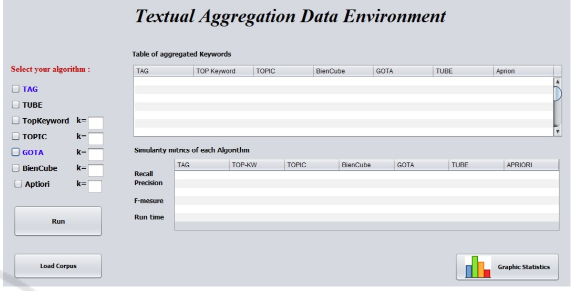
Number of selected documents : 600

Number of all selected Words : 800000

Number of selected Keywords : 2182

Time of corpus generation (ms): 5.3254

Figure 2: The Text Pre-processor and Keywords Extraction component interface.


Textual Aggregation Data Environment

Select your algorithm :

- □ TAG
- □ TUBE
- □ TopKeyword k=
- □ TOPIC k=
- □ GOTA k=
- □ BienCube k=
- □ Aptiori k=

Table of aggregated Keywords

TAG	TOP Keyword	TOPIC	BienCube	GOTA	TUBE	Apriori

Simularity mitrics of each Algorithm

	TAG	TOP-KW	TOPIC	BienCube	GOTA	TUBE	APRIORI
Recall							
Precision							
F-mesure							
Run time							

Figure 3: Keywords Aggregation component interface.

The GUI consists of two components: the first one is devoted to the preprocessing and keywords extraction and the second one is for Keywords aggregation. The Text Pre-processor and Keywords extraction components allow the user to create the *Documents* x *keywords* matrix based on Microsoft Academic Search web site (MAS) as shown in Figure 2. This interface gives users different possibilities to choose and configure the different parameters such as *Threshold* level and select the type of corpus (computer science, medicine, chemistry or all field of study). For the second interface which is devoted for Keywords Aggregation, it allows the user 1- to run, tests and compare the results obtained by the different implemented algorithms. 2- to visualize the aggregated keywords obtained by the different keywords aggregation approaches. 3- to compute different statistics for different approaches such as recall, precision, F-measure and run time, and save the different obtained results in various format *.xls*, *.txt* or *.doc* . 4- to change the corpus and run the Text Preprocessor besides Keywords extraction components to load an other *Documents* x *keywords* Matrix, as shown in Figure 3.

4 RESULTS AND USAGES

4.1 Test and Results

In this subsection, we present an example to show how OLAP-TAS has been used. We compiled two real corpora, the first is from the *IIT* conference [2] (conference and workshop papers) from the years 2008 to 2012. It consists of 600 papers ranging from 7 to 8 pages in IEEE format, including tables and figures. The keywords are extracted from the full words according to the Microsoft Academia Search [3] keywords. The second corpus is used by many authors to test their works such as (Sebastiani, 2002) (Moschitti, 2003) (Moschitti and Basili, 2004), this corpus is called Ohsumed collection [4], it includes medical abstracts from the MeSH (Medical Subject Headings) [5], it contains 20,000 documents. In our case we selected 13,000 medical abstracts to test the performance of the implemented algorithm in our OLAP-TAS. For the evaluation task, many types of measures

[2] http://www.it-innovations.ae

[3] academic.research.microsoft.com/

[4] ftp://medir.ohsu.edu/pub/ohsumed

[5] http://www.ncbi.nlm.nih.gov/mesh/

have been proposed to evaluate keywords aggregation approaches, the majority of them insist on three measures, which are known as recall, precision, and F-measure. these measures are defined as fallows: The recall is the ratio of the number of documents to the total number of retrieved documents.

$$Recall = \frac{|\{RelevantDoc\} \cap \{RetrievedDoc\}|}{|\{RelevantDoc\}|} \quad (1)$$

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

$$Precision = \frac{|\{RelevantDoc\} \cap \{RetrievedDoc\}|}{|\{RetrievedDoc\}|} \quad (2)$$

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

To show the kind of results and statistics obtained by OLAP-TAS after the execution, we take the first corpus as an example to illustrate the different graphs obtained for different algorithms in Figures 4, 5, 6 and 7.

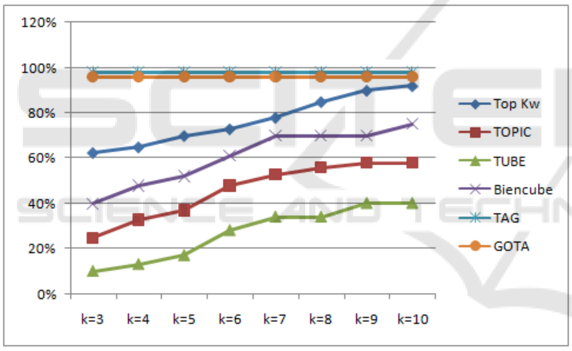


Figure 4: Comparaison of the Recall.

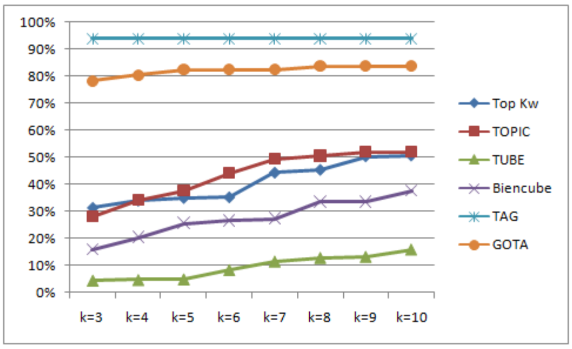


Figure 5: Comparaison of the Precision.

4.2 Uses of OLAP-TAS

In this section we will illustrate the use of the developed tool in both education and research.

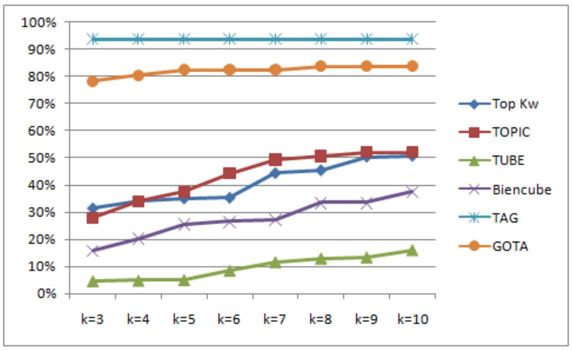


Figure 6: Comparaison of the F-measur.

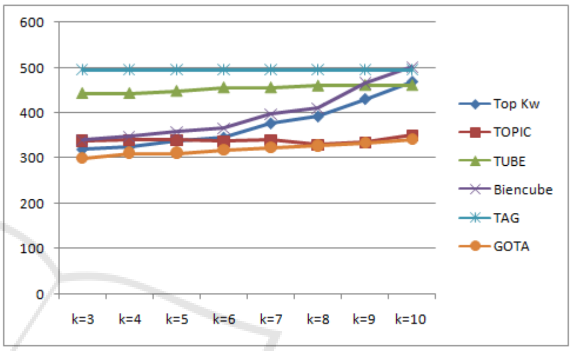


Figure 7: Comparaison of the Runtime.

Education: OLAP-TAS is a visual tool that instructors can use to help their students understand the basic concepts and the algorithms they face during their study. For example, it can be used to teach the students how the k-bisecting clustering algorithm based on the Jensen-Shannon divergence for the probability distribution works (Wartena and Brussee, 2008). As well as the $TF * IDF$ and their variation in Top-keyword (Ravat et al., 2008) and Biencube (Bringay et al., 2011). It can also help students to understand how to use graphs for textual by the selection of cycles in TAG (Bouakkaz et al., 2014) and the use of Google similarity distance (Cilibrasi and Vitanyi, 2007). In addition it shows the students how the recall, precision and F-measure change their values according to number of aggregated keywords k introduced by the user. Instructors may ask their students to do experiments with a real corpus using OLAP-TAS, write applications that use the Java classes, extend an existing approaches, or contribute in implementing a new algorithm to integrate in OLAP-TAS.

Research: OLAP-TAS contains implementations for several algorithms and approaches that solve common problems, such as textual aggregation in an OLAP context. It also comes with two corpora and annotated datasets. The implementation of other algorithms as well as other corpora, can be integrated into the platform. This makes it a good resource for researchers to build systems and conduct experiments. OLAP-TAS was successfully used in several research projects as shown in (Bouakkaz et al., 2014).

5 CONCLUSIONS

In this paper a system for textual aggregation in text OLAP (OLAP-TAS) has been described. The software assists the user to discover the main aggregated keywords that best represent in a document collection. It is important to note that each approach is coded in a separate Java class to allow users to extend it or export it to another system. The use of OLAP-TAS reduces the amount of repeated code; it simplifies common tasks, and provides a graphical interface for textual aggregation approaches without requiring the knowledge in Java programming language.

REFERENCES

Archetti, F. and Campanelli, P. (2006). A hierarchical document clustering environment based on the induced bisecting k-means. *International Conference on Database and Expert Systems Applications*, pages 257–269.

Blei, D. and Andrew, Y. (2003). Latent dirichlet allocation. *Journal of Machine Learning Research*, 42:993–1022.

Bouakkaz, M., Loudcher, S., and Ouinten, Y. (2014). Automatic textual aggregation approach of scientific articles in olap context. *10th International Conference on Innovations in Information Technology*.

Bouakkaz, M., Loudcher, S., and Ouiten, Y. (2015). Gota: Using the google similarity distance for olap textual aggregation. *17th International Conference on Enterprise Information Systems (ICEIS)*.

Bringay, S., Laurent, A., and Poncelet, P. (2011). Towards an on-line analysis of tweets processing. *Database and Expert Systems Applications*, pages 154–161.

Cilibrasi, R. and Vitanyi, P. (2007). The google similarity distance. *IEEE Transactions on Knowledge and Data Engineering*, pages 370–383.

Hady, W., Ecpeng, L., and HweeHua, P. (2007). Tube (text-cube) for discovering documentary evidence of associations among entities. *Symposium on Applied Computing*, pages 824–828.

Kohomban, U. and Lee, W. S. (2007). Optimizing classifier performance in word sense disambiguation by redefining sense classes. *International Joint Conference on Artificial Intelligence*, pages 1635–1640.

Mihalcea, R. and Tarau, P. (2004). Textrank: Bringing order into texts. *Empirical Methods in Natural Language Processing*, pages 26–31.

Moschitti, A. (2003). Natural language processing and text categorization: a study on the reciprocal beneficial interactions. *PhD thesis, University of Rome Tor Vergata, Rome, Italy*, pages 34–47.

Moschitti, A. and Basili, R. (2004). Complex linguistic features for text classification: a comprehensive study. *The 26th European Conference on Information Retrieval Research*, pages 34–47.

Oukid, L., Asfari, O., and Bentayeb, F. (2013). Cxt-cube: Contextual text cube model and aggregation operator for text olap. *International Workshop On Data Warehousing and OLAP*, pages 56–61.

Poudat, C., Cleuziou, G., and Clavier, V. (2006). Cleuziou g., and clavier v., categorisation de textes en domaines et genres. complementarite des indexations lexicale et morpho syntaxique. *Lexique et morphosyntaxe en RI*, 9:61–76.

Ravat, F., Teste, O., and Tournier, R. (2007). Olap aggregation function for textual data warehouse. *In International Conference on Enterprise Information Systems*, pages 151–156.

Ravat, F., Teste, O., and Tournier, R. (2008). Top keyword extraction method for olap document. *In International Conference on Data Warehousing and Knowledge Discovery*, pages 257–269.

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Computing Surveys*, pages 34–47.

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

Zhang, D., Zhai, C., and Han, J. (2009). Topic cube: Topic modeling for olap on multidimensional text databases. *International Conference on Data Mining*, pages 1124–1135.