Appl Intell
DOI 10.1007/s10489-017-1050-9

Efficiently mining frequent itemsets applied for textual aggregation

Mustapha Bouakkaz[1] · **Youcef Ouinten**[1] · **Sabine Loudcher**[2] · **Philippe Fournier-Viger**[3]

Abstract Text mining approaches are commonly used to discover relevant information and relationships in huge amounts of text data. The term data mining refers to methods for analyzing data with the objective of finding patterns that aggregate the main properties of the data. The merger between the data mining approaches and on-line analytical processing (OLAP) tools allows us to refine techniques used in textual aggregation. In this paper, we propose a novel aggregation function for textual data based on the discovery of frequent closed patterns in a generated documents/keywords matrix. Our contribution aims at using a data mining technique, mainly a closed pattern mining algorithm, to aggregate keywords. An experimental study on a real corpus of more than 700 scientific papers collected on Microsoft Academic Search shows that the proposed algorithm largely outperforms four state-of-the-art textual aggregation methods in terms of recall, precision, F-measure and runtime.

Keywords Data mining · Closed keywords · Textual aggregation · OLAP

1 Introduction

The field of data mining offers a set of techniques and intelligent tools to address data exploration challenges [1]. Data mining, also called Knowledge Discovery in Databases (KDD), is "the non trivial extraction of implicit, previously unknown, and potentially useful information from large amount of data" [2]. Data mining [3, 4] is an interdisciplinary field that draws on database theory, machine learning, and statistics to provide powerful data exploration techniques. Numerous data mining techniques have been designed to discover various types of patterns in data, including rules and patterns summarizing the main properties of the data. In recent years, the design of data mining techniques has become a major research area, as well as their use to solve practical data analysis problems in real-world applications [5]. Generally, the goal of data mining is to extract useful and relevant knowledge from data.

In recent decades, the amount of textual information available and stored electronically has been growing at a staggering rate. The best example of this growth is enterprise documents, which are estimated to contain 80% of the useful data that is not exploited by decision makers, given the lack of online analytical processing (OLAP) approaches suitable for textual content [3]. Hence, given the typically large amount of enterprise documents, an important challenge is to develop techniques to efficiently aggregate interesting patterns, trends and information of interest to users [6]. Text mining, also known as Intelligent Text Analysis, Text Data Mining or Knowledge-Discovery in Text, generally refers to the process of extracting interesting and

✉ Mustapha Bouakkaz
m.bouakkaz@lagh-univ.dz

Youcef Ouinten
ouinteny@lagh-univ.dz

Sabine Loudcher
sabine.loudcher@univ-lyon2.fr

Philippe Fournier-Viger
philfv8@yahoo.com

[1] LIM Laboratory, Laghouat University, Laghouat, Algeria

[2] ERIC Laboratory, Lyon 2 University, Lyon, France

[3] Harbin Institute of Technology Shenzhen, Shenzhen, China

Published online: 11 August 2017

non-trivial information and knowledge from unstructured text [7]. It is therefore crucial to combine OLAP with text mining techniques [8] to efficiently provide relevant aggregated information to users. Unstructured data poses new challenges for data analysis compared to structured data found in traditional relational databases [9]. Text mining can work with unstructured or semi-structured data sets such as emails, full-text documents and web pages [10]. A review of general approaches for text mining and knowledge discovery can be found in [11]. Text mining shares many characteristics with classical data mining, but differs in many ways [12].

Text mining is a multidisciplinary field, involving information retrieval, text analysis, information extraction, categorization, visualization, database technology, machine learning, and data mining [13]. Text mining approaches first process unstructured text documents using natural language processing techniques to extract and aggregate keywords, called items. Then, classical data mining techniques are applied on the extracted data (keywords) to discover aggregated patterns. Starting with a collection of documents, a text mining process retrieves a particular document and preprocesses it by checking its format and character sets. Then, a text mining approach goes through a text analysis phase.

In recent years, due to the rapid increase in text data availability and the lack of OLAP approaches to handle this type of data, there is a growing need to use data mining techniques to extract useful knowledge from this data such as aggregated keywords. Furthermore, the large number of unstructured documents stored electronically in document warehouses or on the Web makes standard OLAP techniques unsuitable for handling large collections of enterprise documents. These limitations of traditional OLAP have led to the recent development of approaches for mining aggregated information such as aggregated keywords from unstructured data [20, 21, 26].

Information aggregation approaches can be used to directly extract knowledge from a text corpus, or to extract representative keywords from a set of documents, which can then be analyzed using traditional data-mining techniques to discover complex patterns [14]. Many text mining methods have been developed to retrieve aggregated information that is useful for users [15]. Most text mining methods use keyword based approaches to construct textual representations of sets of documents.

In this paper we thus address the challenges of proposing a new textual aggregation function for the OLAP context based on data mining techniques. The contribution of this paper is a novel measure named FP-COTA based on the discovery of frequent closed patterns in a generated document keyword/matrix. An experimental study on a real corpus of more than 700 scientific papers collected on Microsoft Academic Search shows that the proposed measure outperforms four state-of-the-art textual aggregation methods in terms of recall, precision and F-measure.

The rest of the paper is organized as follows: Section 2 reviews related work related to the combination of OLAP and data mining. Section 3 presents the proposed method, named FP-COTA. Section 4 presents an experimental study, which compares the performance of the proposed method with four state-of-the-art approaches on a real corpus. Finally, Section 5 draws conclusions and discusses future work.

2 Combining OLAP and data mining

OLAP and data mining are used to solve different kinds of analytic problems. On one hand, OLAP operators are powerful mechanisms for organizing and structuring data to allow exploration and navigation of aggregated data. On the other hand, data mining techniques are known for their descriptive and predictive power, to discover knowledge in data. Thus, OLAP provides aggregated data and performs complex calculations while data mining discovers hidden patterns in data. Hence, OLAP and data mining are complementary, and combining them can result in a more elaborated analysis. In the context of data warehouses and OLAP, some data mining techniques can be used as aggregation operators. Thus, many studies are now developping more complex operators to take advantage of the data analysis capabilities provided by data mining [16, 17]. This paper goes beyond these proposals by proposing a novel aggregation approach based on the extraction of frequent closed patterns.

This section reviews existing aggregation approaches. It classifies existing approaches into four categories: approaches based on linguistic knowledge, external knowledge, graph and statistical information. Characteristics of these approaches are discussed in the following paragraphs. To our best knowledge, no previous studies have considered using frequent patterns for textual aggregation.

Approaches based on linguistic knowledge view a corpus as the set of words (a vocabulary) appearing in its documents, which may results in ambiguities. To overcome this obstacle, techniques based on lexical knowledge and syntactic knowledge have been introduced. Poudat et al. [18] and Bouakkaz et al. [19] proposed a classification of textual documents based on scientific lexical variables of discourse. Among these lexical variables, they chose nouns rather than adverbs, verbs or adjectives, because nouns are more likely to emphasize scientific concepts.

Approaches based on the use of external knowledge select specific keywords that represent a given domain. These approaches often use knowledge models such as ontologies. Ravat et al. proposed an aggregation function that takes as input a set of keywords extracted from a corpus of documents and outputs a set of aggregated keywords [20]. They assumed that both the ontology and the corpus of documents belong to the same domain. Oukid et al. [21] proposed an aggregation operator named Orank (OLAP rank), which aggregates a set of documents by ranking them in a descending order using a vector space representation. Subhabrata et al. in [22] proposed a textual aggregation model using an ontology. They proposed an approach to construct keywords Ontology Tree [23].

Approaches based on graphs use keywords to construct a keyword graph, where each node represents a keyword. Keywords are obtained by a process of pre-processing and candidate selection. In a keyword graph, an edge indicates the strength or relatedness (or semantic relatedness) of the two keywords that it connects. After building a graph, different types of keyword ranking approaches can be applied. The first proposed approach is called TextRank [24]. It builds a graph where edges represent co-occurrence relations between keywords in the corpus. The assumption of this approach is that if a keyword is linked to many other keywords, then it is considered as important. An edge between two keywords can be interpreted as a measure of their semantic relatedness. TextRank applies the PageRank algorithm to obtain the PageRank score for each pair of terms to rank candidates. TextRank tends to extract high-frequency terms as keywords because these terms have more opportunities to be linked with other terms and hence obtain higher PageRank scores. The co-occurrences relationship used to build TextRank's term graph can be seen as an approximation of the semantic relationships between words. As a consequence, TextRank may connect semantically unrelated words, and may introduce noise, which may negatively influence extraction performance. An alternative approach to alleviate the vocabulary gap is to use latent topic models. Latent topic models learn topics from a collection of documents. Using a topic model, both documents and terms can be represented as distributions over latent topics. The semantic relatedness between a term and a document can be estimated using the similarities of their topic distributions. Similarity scores can be used as ranking criteria for keyword extraction [25]. Bouakkaz et al. [26] proposed a method, which performs aggregation of keywords in documents based on the construction of a graph using affinities between keywords, and the identification of cycles in the graph. This process selects the main aggregated keywords from a set of terms representing a corpus. The aggregation approach proposed in this work is called TAG (Textual Aggregation by Graph). It takes as input the set of all extracted terms from a corpus, and outputs an ordered set containing the most aggregated keywords. The process of aggregation goes through three steps: (1) Extraction of keywords with their frequencies, (2) Construction of an affinity matrix and affinity graph, and (3) Cycle identification and aggregated keywords selection.

Approaches based on statistical methods consider occurrence frequencies of terms and correlation between terms. The authors of [28] proposed the LSA (Latent Semantic Analysis) method in which the corpus is represented by a matrix where each row represents a document and each column represent a keyword. Each matrix element stores the number of occurrences of a word in a document. By then performing decomposition and reduction, this method identifies a set of keywords that represents the corpus. The authors of [29] proposed an approach called TUBE (Text-cUBE) to discover associations between entities. The model adapts the concept of data cubes designed for relational databases to textual data, where cells contain keywords, and an interestingness value is attached to each keyword. The authors of [30] proposed two aggregation functions. The first one is based on a new adaptive measure based on Tf.Idf which takes into account hierarchies associated to dimensions. The second one is built dynamically and is based on clustering. The authors of [31] used the k-bisecting clustering algorithm based on the Jensen-Shannon divergence of probability distributions [32]. Their method starts by creating two clusters containing the two elements that are the most far apart as seed clusters. Each other element is then assigned to the cluster having the closest seed. Once all elements have been assigned to clusters, the centers of both clusters are computed. The new centers are then used as new seeds for finding two new clusters. This process is repeated until each of the two new centers converge up to a precision lower than some predefined threshold value. Then, if the diameter of a cluster is larger than a specified threshold value, the whole procedure is applied recursively to that cluster to divide in into two clusters. The authors of [33] proposed a second aggregation function called TOP-Keywords to aggregate keywords. It computes the frequencies of terms using the $Tf.Idf$ function, and then selects the first k most frequent terms. Jingxuan et al. in [34] proposed a framework for different multi-document aggregation tasks using submodular functions based on the term coverage and the textual-unit similarity which can be efficiently optimized through the improved greedy algorithm. They show that four known aggregation tasks, including generic, query-focused, update, and comparative aggregation, can be modeled as different variations derived from the proposed framework. The authors of [35] proposed the C-Value algorithm, which ranks potential keywords by using the length of the phrases containing the keywords, and their frequencies. The authors of [36] proposed a technique for extracting sentences summarizing a set of documents by considering the weights of the sentences and the documents. The authors of [27] proposed an approach called top-bottom to perform document aggregation using texture features that are extracted

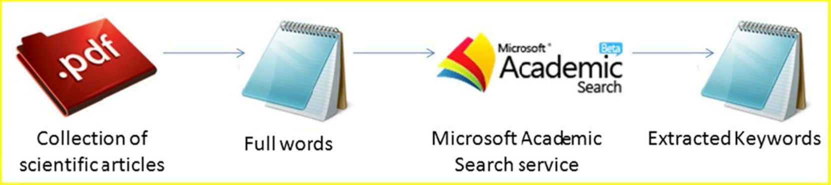
.pdf
Collection of scientific articles
Full words
Microsoft Academic Search Beta
Microsoft Academic Search service
Extracted Keywords

Fig. 1 Steps of keyword extraction

from the specified/selected documents. A mask of suitable size is used to aggregate textural features, and statistical parameters are captured as blocks in document images. Four textural features that are extracted from masks using the gray level co-occurrence matrix. Furthermore, two statistical parameters extracted from corresponding masks are the modal and median pixel values.

3 The proposed method: FP-COTA

This paper aims at creating a suitable environment for the online analysis of documents by taking into account textual data. In Text OLAP, a measure can be textual, i.e. it may return a list of keywords as result. If a user wants to obtain a more aggregated view of the data, he can then apply a roll-up operation. Implementing this operation requires an adapted aggregation function that performs text data aggregation. The following paragraphs introduce the proposed approach step-by-step by discussing its design and implementation. The proposed approach is named FP-COTA. It performs three main operations: (1) extracting keywords with their frequencies; (2) generating frequent closed patterns; (3) selecting k aggregated keywords.

3.1 Extracting keywords

The first step of the proposed approach is to select a set of terms T by cleaning stop words, applying lemmatization, and then selecting the most significant terms. There are different ways of selecting terms. In this work, the weight (frequency) of a term is used as it assesses the importance degree of the term in a document. These weights are defined as follows:

$$\forall t_i \in T,\ w_i = -\frac{tf_i}{tf^i} \qquad (1)$$

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

3.2 Discovering frequent closed patterns

The second step of the proposed approach is to extract frequent closed patterns. The concept of closed pattern was initially proposed for market basket analysis to analyze customer transactions [37]. In the context of this paper, a closed pattern is a set of keywords such that there does not exist a superset of keywords having the same support (occurrence frequency) in the Documents/Keywords matrix. In other words, a closed pattern is a set of keywords that always appear together in documents. Thus the support of any pattern is the same as the support of the smallest closed pattern containing it.

In the proposed approach, frequent closed patterns are mined from the Documents/Keywords matrix by performing two sub-steps: 1) building an FP-Tree from the matrix, and 2) retrieving frequent patterns from the FP-Tree, and then filtering out all non-closed patterns. In the method implemented by Grahne [37], all frequent patterns are mined using the FP-Growth algorithm and then stored in a T-Tree structure (Total Support Tree), which also stores the support of all frequent patterns. Then, a post-processing step is performed to only output the closed patterns stored in the T-Tree. To avoid storing the set of all frequent patterns and performing this post-processing step, we adopt the implementation called FPClose provided in the SPMF library [38], where the T-Tree contains only the closed frequent patterns, and which provides fast access to the set of closed frequent patterns.

3.3 Aggregated keywords selection

The third step of the proposed approach is aggregated keyword selection. All the frequent closed patterns discovered in the previous step are ordered by their support. The system lets the user choose the size of the largest pattern in terms of number of items K to be selected. Patterns representing the most representative keywords in the corpus (having the highest value of support) are selected.

4 Experimental study

To evaluate the proposed approach, the authors of this paper have compiled a corpus from the *IIT* conference[1] (conference and workshop papers) from the years 2008 to 2014. It consists of 700 papers with a length of 7 to 8 pages in IEEE format, including tables and figures. Keywords have

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

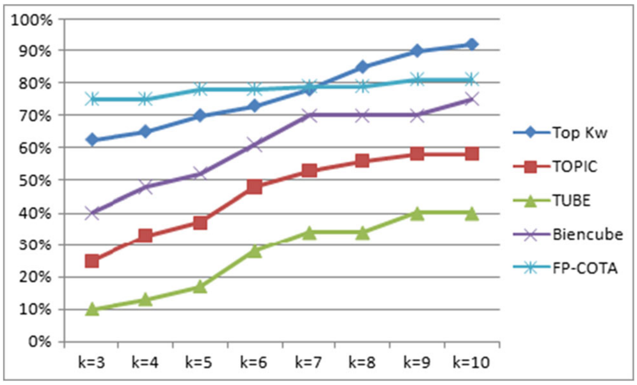

Fig. 2 Recall of different approaches

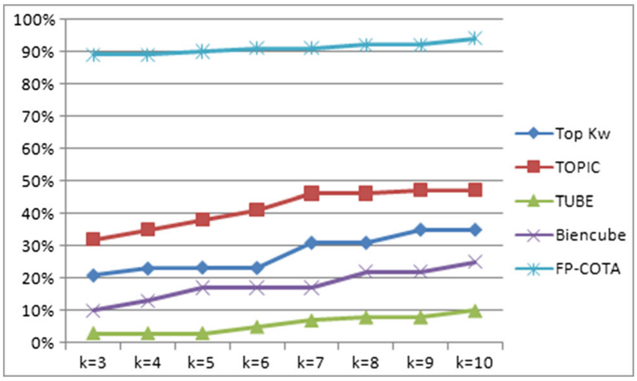

Fig. 3 Precision of different approaches

been extracted from the whole list of words using Microsoft Academic Search[2] keywords.

The keywords extraction function is based on the Microsoft Academic Search web site (MAS) as shown in Fig. 1. MAS classifies scientific articles according to fifteen scientific fields by extracting scientific keywords from articles and ordering them according to their frequencies. Among the lists of keywords produced by MAS, the 2000 most frequent keywords were selected from each field. Extracting keywords from the corpus is then done according to these lists of keywords for each field. The output of this process is the two fold matrix of *Documents* x *Keywords*, which is used to compare the proposed approach with other textual aggregation approaches.

Several measures have been proposed to evaluate keyword aggregation methods [40–42]. But the most frequently used are the recall, precision, and F-measure [39].

The recall is the ratio of the number of documents to the total number of retrieved documents [39].

$$Recall = \frac{\{RelevantDoc\} \cap \{RetrievedDoc\}|}{|\{RelevantDoc\}|} \tag{2}$$

The precision is the ratio of the number of relevant documents to the total number of retrieved documents [39].

$$Precision = \frac{\{RelevantDoc\} \cap \{RetrievedDoc\}|}{|\{RetrievedDoc\}|} \tag{3}$$

The F-measure or balanced F-score, combines the precision and recall measures. It is defined as the harmonic mean of the precision and recall [39].

4.1 Results

This sections presents an empirical study to evaluate the proposed aggregated keyword function using two real corpora. Its performance is compared with the performance of Tube [29] Biencube [30] Topic [31] and TopKeywords [33].

Experiments were performed on a PC running the Microsoft Windows 7 Edition operating system, equipped with a 2.62 GHz Pentium Dual-core CPU, 4.0 GB main memory, and a 300 GB hard disk. To test and compare the different approaches, the corpus described in the previous section was used, which contains 700 articles, 950.000 terms and 3.214 extracted keywords.

To compare the methods, the recall, precision, F-measure and runtime were measured for different values of K. A comparison of the complexity of the five algorithms was also done. Results from the experiments are summarized in Figs. 2, 3, 4 and 5. It can be observed that the proposed approach yield the highest values in terms of recall, precision and F-measure.

For instance, for $K = 3$, the proposed FP-COTA approach has a recall of 75%, while Topkeyword, TOPIC, BienCube and TuBE obtain 63%, 25%, 40% and 10%, respectively. In terms of precision, FP-COTA reaches 89% while Topkeyword, TOPIC, BienCube and TuBE obtain 21%, 32%, 10% and 3%, respectively. In terms of F-measure, the designed approach obtains 81%, while Topkeyword, TOPIC, BienCube and TuBE have 31%, 28%, 16% and 5%, respectively. For $K = 10$, the proposed approach has a recall of 81%, while Top- keyword, TOPIC,

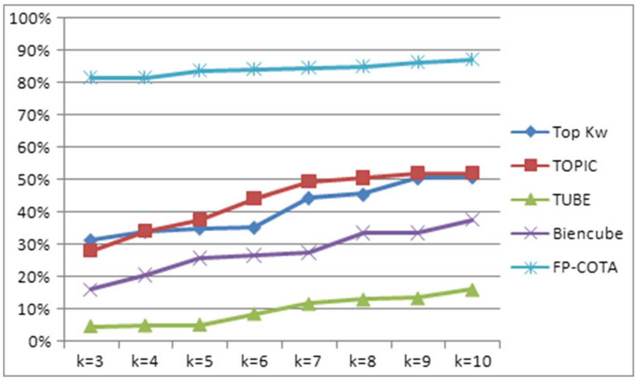

Fig. 4 F-measure of different approaches

[2] academic.research.microsoft.com/

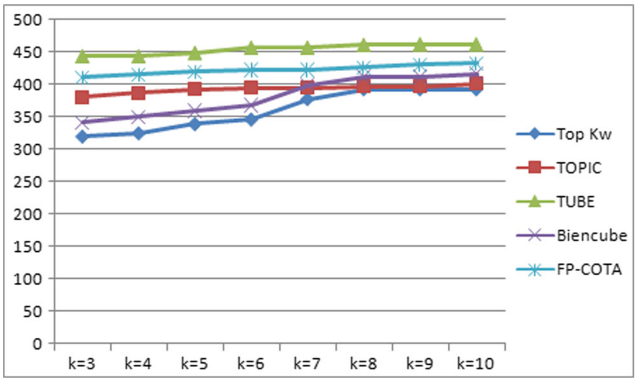

Fig. 5 Runtime of different approaches (ms)

BienCube and TuBE gets 92%, 58%, 75% and 40%, respectively. The proposed approach obtains a precision of 94%, while Topkeyword, TOPIC, BienCube and TuBE obtain 35%, 47%, 25%, respectively. Lastly, in terms of F-measure, the proposed approach obtains 87%, while Topkeyword, TOPIC, BienCube and TuBE obtain 51%, 52%, 38%, respectively.

In Fig. 5 it can be observed that there is a considerable difference between the five compared approaches. The reason is that they do not have the same time complexity. The proposed FP-COTA approach applies the FPClose algorithm as a sub-step which has a complexity of $O(X)$, where X is the number of closed patterns. The number of closed patterns is in the worst case $2^Y - 1$ where Y is the number of keywords in the longest document [38]. But in practice the number of patterns can be much less depending on how the minsup threshold is set. Topkeyword and BienCube have a complexity of $O(N)$ [30, 33]. On the other hand TOPIC is based on the k-bisecting clustering which has a complexity of $O((k1)kN)$. where K is the number of clusters and N the number of terms [31]. TUBE has a complexity of $O(N^2)$ [29]. Thus, although the complexity of the proposed approach may seem high when compared to simpler approaches, it can be seen as an interesting trade-off for higher recall, precision and F-measure.

Overall, based on the results presented in Figs. 2, 3, 4 and 5, it can be seen that the designed approach outperforms other approaches in terms of recall, precision and F-measure, and has good performance in terms of runtime.

5 Conclusions

This paper has presented a novel textual aggregation function named FP- COTA based on the use of frequent closed itemset mining to generate aggregated keywords for text OLAP. Closed patterns proved to be a suitable alternative to clustering and classical approaches used in previous work due to their flexibility and low complexity. Moreover, pattern mining has an advantage over classical OLAP techniques, as it better supports generating empirical solutions for textual aggregation. To evaluate the proposed approach, a real corpus was collected, and the performance of the approach was compared to four state-of-the-art methods for textual aggregation. Results have shown that the proposed approach outperforms the compared approach in terms of recall, precision, F-measure and runtime. For future work, we will consider introducing the semantic aspect of keywords in FP-COTA as well as using other corpora.

References

1. Frawley W, Piatetsky-Shapiro G, Matheus C (1992) Knowledge discovery in databases: an overview. AI Magazine, fall 1992, pp 213–228
2. Palmerini P (2004) On performance of data mining: from algorithms to management systems for data exploration (Doctoral dissertation, PhD. Thesis: TD-2004-2, Universita CaFos- cari di Venezia)
3. Han J, Fu Y (1996) Attribute-oriented induction in data mining, advances in knowledge discovery and data mining. AAAI Press/The MIT Press, pp 399–421
4. Han J, Kamber M (2001) Data mining: concepts and techniques. Morgan Kaufman, San Mateo. ISBN: 1-55860489-8
5. Chen SY, Liu X (2005) Data mining from 1994 to 2004: an application-oriented review. Int J Business Intell Data Min 1(1):4–11
6. Baeza-Yates R, Moffat A, Navarro G (2002) Searching large text collections. Handbook of massive data sets, pp 195–243. ISBN:1-4020-0489-3
7. Navathe S, Ramez E (2000) Fundamentals of database systems. Pearson Education, Singapore
8. Wu S-T, Li Y, Xu Y (2006) Deploying approaches for pattern refinement in text mining. In: Proceedings of the sixth IEEE international conference on data mining, pp 1157–1161
9. Gupta V, Gurpreet SL (2009) A survey of text mining techniques and applications. J Emerg Technol Web Intell 1(1):60–76
10. Delgado M, Martín-Bautista MJ, Sánchez D, Vila MA (2002) Mining text data: special features and patterns. In: Proceedings of EPS exploratory workshop on pattern detection and discovery in data mining, London
11. Kodratoff Y (1999) Knowledge discovery in texts: a definition and applications. In: Proceedings of the 11th international symposium on foundations of intelligent systems, pp 16–29
12. Tan A-H (1999) Text mining: the state of the art and the challenges. In: Proceedings of the PAKDD workshop on knowledge discovery from advanced databases, pp 65–70
13. Nasukawa T, Nagano T (2001) Text analysis and knowledge mining system. IBM Syst J 40(4):967–984
14. Mooney RJ, Bunescu R (2005) Mining knowledge from text using information extraction. ACM SIGKDD Explor Newsl 7(1):3–10
15. Leopold E, Kindermann J (2002) Text categorization with support vector machines. How represent texts in input space? Mach Learn 46:423–444
16. Xu X, Mete M, Yuruk N (2005) Mining concept associations for knowledge discovery in large textual databases. In: Proceedings of the 2005 ACM symposium on applied computing. ACM, pp 549–550. ISO 690
17. Mahgoub H (2006) Mining Association rules from unstructured documents. World Acad Sci Eng Technol 20(1):1–6

18. Poudat C, Cleuziou G, Clavier V (2006) Catgorisation de textes en domaines et genres. Doc Numrique 9(1):61–76
19. Bouakkaz M, Loudcher S, Ouinten Y (2016) OLAP textual aggregation approach using the Google similarity distance. Int J Bus Intell Data Min 11(1):31–48
20. Ravat F, Teste O, Tournier R (2007) OLAP aggregation function for textual data warehouse. In: International conference on enterprise information systems, pp 151–156
21. Oukid L, Asfari O, Bentayeb F, Benblidia N, Boussaid O (2013) CXT-cube: contextual text cube model and aggregation operator for text OLAP. In: Proceedings of the sixteenth international workshop on Data warehousing and OLAP. ACM, pp 27–32
22. Mukherjee S, Joshi S (2014) Author-specific sentiment aggregation for polarity prediction of reviews. In: Ninth international conference on language resources and evaluation. ELRA, pp 3092–3099
23. Ravi K, Ravi V (2015) A survey on opinion mining and sentiment analysis: tasks, approaches and applications. Knowl-Based Syst 89:14–46
24. Mihalcea R, Tarau P (2004) TextRank: bringing order into texts. Association for Computational Linguistics
25. Blei DM, Lafferty JD (2006) Dynamic topic models. In: Proceedings of the 23rd international conference on Machine learning. ACM, pp 113–120
26. Bouakkaz M, Loudcher S, Ouinten Y (2014) Automatic textual aggregation approach of scientific articles in OLAP context. In: 2014 10th international conference on innovations in information technology (INNOVATIONS). IEEE, pp 30–35
27. Oyedotun OK, Khashman A (2016) Document segmentation using textural features summarization and feedforward neural network. Appl Intell 45(1):198–212
28. Hossain MM, Prybutok VR (2016) Towards developing a business performance management model using causal latent semantic analysis. Int J Bus Perform Manag 17(2):161–183
29. Lauw HW, Lim EP, Pang H (2007) Discovering documentary evidence of associations among entities. In: Proceedings of the 2007 ACM symposium on applied computing. ACM, pp 824–828
30. Bringay S, Bchet N, Bouillot F, Poncelet P, Roche M, Teisseire M (2011) Towards an on-line analysis of tweets processing. In: Database and expert systems applications. Springer, Berlin, pp 154–161
31. Wartena C, Brussee R (2008) Topic detection by clustering keywords. In: 19th international workshop on database and expert systems application. DEXA'08. IEEE, pp 54–58
32. Fuglede B, Topsoe F (2004) Jensen-Shannon divergence and Hilbert space embedding. In: IEEE international symposium on information theory, pp 31–31
33. Ravat F, Teste O, Tournier R, Zurfluh G (2008) Top Keyword: an aggregation function for textual document OLAP. In: Data warehousing and knowledge discovery. Springer, Berlin, pp 55–64
34. Li J, Li L, Li T (2012) Multi-document summarization via submodularity. Appl Intell 37(3):420–430
35. Frantzi K, Ananiadou S, Mima H (2000) Automatic recognition of multi-word terms: the c-value/nc-value method. Int J Digit Libr 3(2):115–130
36. El-Ghannam F, El-Shishtawy T (2014) Multi-topic multi-document summarizer. arXiv preprint arXiv:1401.0640
37. Grahne G, Zhu J (2005) Fast algorithms for frequent itemset mining using fp-trees. IEEE Trans Knowl Data Eng 17(10):1347–1362
38. Fournier-Viger P, Gomariz A, Gueniche T, Soltani A, Wu CW, Tseng VS (2014) SPMF: a Java open-source pattern mining library. J Mach Learn Res 15(1):3389–3393
39. Wang T, Chen P, Simovici D (2016) A new evaluation measure using compression dissimilarity on text summarization. Appl Intell 45(1):127–134
40. Sutcliffe T (1992) Measuring the informativeness of a retrieval process. In: Proceedings of SIGIR, pp 23–36
41. Jones K, Willett P (1997) Readings in information retrieval. Morgan Kaufmann, San Mateo
42. Trec: common evaluation measures. The twenty-second Text REtrieval conference. http://trec.nist.gov/pubs/trec22/trec2015.html (2015)