Filomat 32:5 (2018), 1973–1982
https://doi.org/10.2298/FIL1805973Z

Published by Faculty of Sciences and Mathematics,
University of Niš, Serbia
Available at: http://www.pmf.ni.ac.rs/filomat

OLAP on Multidimensional Text Databases: Topic Network Cube and its Applications

Zhiyuan Zhang[a], Hong Wang[a], Xingjie Feng[a]

[a] *School of computer science & technology, Civil Aviation University of China, Tianjin, China*

Abstract. Multidimensional text data contains both structured attributes and unstructured text. Unlike the traditional numerical data, it is not straightforward to apply online analytical processing on multidimensional text data. Although some OLAP methods such as topic cube have been proposed in order to effectively utilize its structured information and valuable text data, these methods cant tell the relations of topic words. Considering that topics usually consist of several subtopics and each subtopic usually contains some topic words, we here use a topic network manner, in which related topic words are connected, to express the complex relations of topics. This paper introduces a new concept of topic network cube to perform OLAP analysis on multidimensional text databases. Firstly, we propose a method called GL-LDA based on Gibbs sampling outputs of Labeled LDA to measure the relations between topic words. Secondly, we give a storage model of topic network cube which can efficiently generate topic network using GL-LDA. Thirdly, we show how to perform OLAP analysis on topic network cube. Experimental results show that we can analyze multidimensional text databases in different granularity easily and effectively using just a few simple SQL statements, and the output network provides rich and useful information of topics.

1. Introduction

Online analytical processing (OLAP) [1] is an important technology to analyze large amount of multidimensional data in different granularity. In recent years, OLAP has been extended from the traditional numerical data to many new domains, such as text data [2, 3], graph data [4, 5] and spatial data [6]. A multidimensional text database refers to data records with both unstructured text data and other structured attributes, with the unstructured text data containing valuable information to be discovered. Text cube [2] is a good attempt to model multidimensional text data for OLAP. The measures of a text cube include term frequency (TF) and inverted index (IV). These two measures can be efficiently computed and one can easily find the most frequent terms or do information retrieval jobs by these two measures.

Topic modeling is a popular way to discover topics hidden in a large amount of documents. Although one can use methods like PLSA [7] (Probabilistic Latent Semantic Analysis) or LDA [8] (Latent Dirichlet Analysis) for untagged documents or use Labeled LDA [9] for tagged documents to find topic terms, it is not straightforward to integrate these algorithms into OLAP, because when OLAP query changes, involved

2010 *Mathematics Subject Classification*. Primary 62-07
Keywords. multidimensional text database; topic network cube; OLAP; text mining; complex network
Received: 26 October 2017; Accepted: 30 January 2018
Communicated by Hari M. Srivastava
Research supported by the National Natural Science Foundation of China (Grant No. 61201414, U1633110), and the Fundamental Research Funds for the Central Universities (Grant No. 3122016D021)
Email address: zy-zhang@cauc.edu.cn (Zhiyuan Zhang)

document sets may also change, therefor we should rerun these algorithms to get a probabilistic topic model, which is very time consuming. Topic cube [3] combines OLAP with probabilistic topic modeling. To partly relieve the time consuming problem discussed above, topic cube proposes a heuristic method to choose a good starting point for iteration in Expectation-Maximization (EM) stage of PLSA model to reduce iteration times. Topic cube extends the traditional data cube to cope with a topic hierarchy and stores probabilistic content measures of text documents learned through a PLSA model. For example, for the topic of "*equipment problems*" during January 1999 in NASA's aviation safety reports (ASRS), it stores a probabilistic term list like "*engine 0.104, pressure 0.029, oil 0.023, checklist 0.022, hydraulic 0.020...*" showing that engine and oil pressure are the main reasons of equipment problems. Since PLSA and LDA are both unsupervised algorithms, it is a human work to capture the exact meaning of the extracted topics. Supervised topic models [9, 10] effectively utilize documents' tag information and can directly get a probabilistic word distribution of a specific topic. Supervised LDA [10] is applicable to single-label documents, and Labeled LDA [9] works better on multi-label documents. Thus for a multidimensional multi-label text database, we can use Labeled LDA instead of PLSA to avoid the topic align issue. Our work is based on Labeled LDA, and we use a graphical view to show the relations of topic words.

Considering that a topic usually consists of several subtopics, it would be a good approach to construct a topic network in which related words are connected to express its complex characteristics. In a topic network, words with strong connections may form into communities, and community structures may reflect the complex relations among words. This graphical form of topic words provides an intuitive and effective information for analyzers. On the other hand, word connections may provide extra information while lack of them may lead to difficulties in understanding. For example, it would be quite confusing to see a word "*first*" appears in the topic of "*equipment problem*", however, when it connects to "*officer*", we will see at once that it is referring to the second pilot. Here we present a new concept of "***topic network cube***" to distinguish it from "*topic cube*" discussed in [3]. A topic network cube can cope with a topic hierarchy and it stores a topic network, i.e. probabilistic measures among words for a specific topic, while "*topic cube*" only stores a list of terms with probabilistic measures.

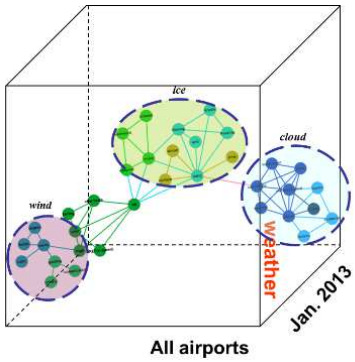


Figure 1: illustration of a topic network cube

Figure 1 is an illustration of a topic network cube. It is about "*weather*" for ASRS reports of all airports in January 2013. There are three subtopics in this network: *wind*, *ice* and *cloud*. Comparing to the previous probabilistic term list, the topic network way clearly has a better visual effect and contains richer information. Query conditions can change on any dimensions, such as time from January 2013 to the 1st quarter of 2013, or topic from weather to physical environment. When query conditions change, the topic network must also change to answer the query. Since the topic network cube is completely different from the numerical data cube, it is not that easy to do multi-scale analysis on it. The main content of this paper is about how to construct a topic network cube, how to store it, and how to do multi-scale or OLAP analysis on it.

2. Probabilistic measurement of topic words

Suppose we have 10 topic words discovered by a topic model, then how to measure their correlation? Should they all be connected? Pointwise Mutual Information (PMI) is a good way of evaluating word correlations. It was first introduced into natural language processing (NLP) by Church [11], and got good performance in word association [11] and synonym recognition task [12]. Smith et al. [13] compute PMI value between top n word pairs for each topic and connect frequent co-occurrence words according to their PMI value to construct a network for each topic. In their work, PMI calculation is a separate part from the topic model, therefore the value is a global correlation between two words, not for a specific topic. If documents scale is not large enough, PMI will be unreliable because of data sparsity [14]. Since the documents scale we processed is not that large, PMI would not be a good solution for us. However, inspired by the idea used in PMI that counting the co-occurrence times of two words in a window, in our previous work, we incorporate PMI into Labeled LDA model, called PL-LDA (Pointwise Labeled LDA) [15] to compute the conditional joint probability of two words for a specific topic, say $p(w_i, w_j|k)$.

Although PL-LDA is a feasible way to compute probabilistic measurements of word pairs for a given topic, it is not applicable to OLAP operations because it costs much more time than Labeled LDA for its larger size corpus. Not only that, when query condition changes, the involved documents set changes too, thus we must rerun PL-LDA to get a new probabilistic measurement of word pairs. Is it possible to compute the conditional joint probabilities with only the original documents? Usually for the outputs of LDA or Labeled LDA, researchers mainly focus on the word-topic matrix ϕ and topic-document matrix θ. As for the gibbs sampling outputs, got little concerned. Actually, the gibbs sampling outputs mark each word a topic, and can be thought as a joint distribution of word W and topic Z. According to this assumption, $p(w_i|z_k)$ can then be evaluated. And based on the assumption that frequently co-occurred words may be highly correlated, $p(w_j|w_i, z_k)$ can also be evaluated. Then by the rule of Bayesian conditional probability, the joint probability distribution $p(w_i, w_j|z_k)$ can be obtained by multiplying them. As this method uses gibbs sampling outputs of Labeled LDA, we call it GL-LDA [16] where G stands for *gibbs sampling*.

The following is an example of gibbs sampling output of an ASRS report (No. 1058455) with topic "*Aircraft Equipment Problem Critical*". Stop words together with symbols and numbers are dropped in preprocess, and are marked with strikethrough. Red italic words are about topic "*Aircraft Equipment Problem Critical*", while blue bold words are about "*general*". Sometimes the word *landing* is put into the *general* topic, sometimes it is put into another, and this is just a reflection of probabilistic annotation.

~~After~~ *takeoff* **got** ~~a~~ *LE FLAP ASYM*. ~~An~~ *Emergency* ~~was~~ *declared* ~~for a~~ *non normal flap configuration* **landing**. ~~An~~ **approach** ~~and~~ *landing* ~~was~~ *made* ~~at the~~ **departure airport**. ~~Over~~ *weight landing* ~~of 6,500~~ *lbs*. ~~was~~ **completed** ~~with a 150~~ *feet VS* ~~or~~ *less per*/**min**. ~~No~~ *evacuation*; ~~no~~ *injuries*; ~~and no~~ *fire*.

According to Bayesian rules, $P(t, s|k) = P(t|k) * P(s|k, t)$, where $P(t|k)$ is the word topic matrix, and is evaluated by,

$$\phi_k^t = \frac{n_k^{(t)} + \beta_t}{\sum_{t'=1}^{V}(n_k^{(t')} + \beta_{t'})}, \tag{1}$$

where $n_k^{(t)}$ is the times that word t is annotated as topic k, β_t is the prior probability of word t, and V is the vocabulary size. When word t is annotated as topic k, other words appearing in its 2L-size window can be thought as co-occurred with it, thus the conditional probability $P(s|k, t)$ can be evaluated by:

$$\phi_{k,t}^s = \frac{n_{k,t}^{(s)} + \beta_s}{\sum_{s'=1}^{V}(n_{k,t}^{(s')} + \beta_{s'})}, \tag{2}$$

where $n_{k,t}^{(s)}$ is the times that word s appears in the 2L-size window of word t annotated as topic k. If $n_{k,t}^{(s)}$ is stored in the computing stage of Labeled LDA, a matrix of K by V^2 is needed, which may be too large for ordinary PC memory. Instead, we store the gibbs sampling outputs in a database, and use its self-join operation to calculate $n_{k,t}^{(s)}$.

Our previous experiments [16] show that if the involved documents number is large, the result of GL-LDA is almost the same as the result of PL-LDA. Furthermore, we only run Labeled LDA once to get its gibbs sampling outputs, and it is all the stuff we need. Therefore, GL-LDA is a better solution for OLAP.

3. Storage model of topic network cube

3.1. Storage model

As an example, the storage model of topic network cube for ASRS reports are illustrated in figure 2. The right part (pink area) is mainly for data storage, and the left part (light blue area) is mainly for query. In the right part, the *Words* table stores the gibbs sampling outputs of Labeled LDA for all documents, including document ID, offset (the position of a word in the document), the word itself, and the topic ID which it is marked. The *WordABK* table is the fact table, which is obtained through self-join of the Words table using the following SQL statement (here we suppose the window size of GL-LDA is 10):

insert into wordABK
select A.documentID, A.word, A.offset, A.topicID, B.word, B.offset, B.topicid from words A
inner join words B on A.documentID=B.DocumentID and abs(A.offset-B.offset) between 1 and 5

The blue thick arrow means that *wordABK* depends on *Words*. The *Documents* table stores all the documents information, the *Dates* table stores the hierarchical date information, the *Locations* table stores the hierarchical location information, and the *topic* table stores the hierarchical topic information. The topic hierarchical tree of ASRS reports in our experiment is shown in figure 3. There are 22 topics in our experiment, and are grouped into 7 categories. The left part is used for query and statistics. The yellow thick arrow shows the dependencies of these tables. For example, when query condition changes, the *SearchDocuments* table will select records from the *Documents* table to meet the conditions.

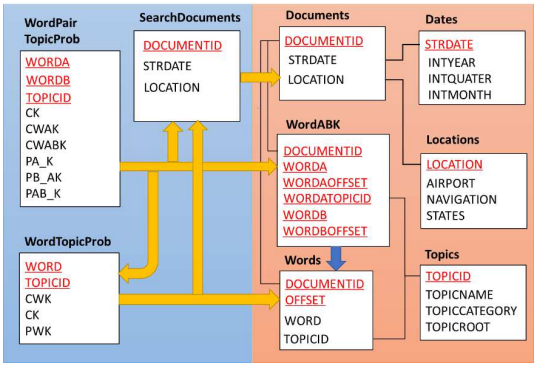


Figure 2: storage model of topic network cube

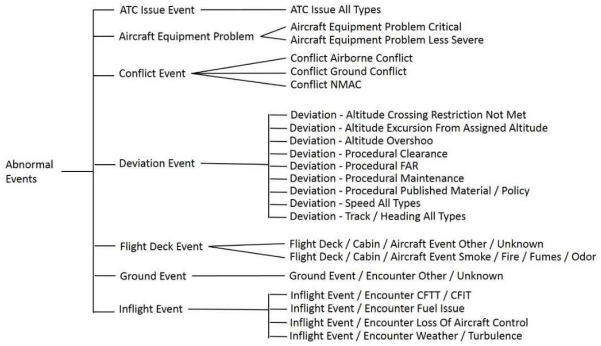


Figure 3: topic hierarchical tree of ASRS reports

3.2. table structures

Table 1 and 2 are the table structures of *WordTopicProb* and *WordPairTopicProb*. Fields with underline are the primary keys. Other tables are easy to understand by their attribute names and are not listed here.

3.3. Topic network construction

A topic network uses a graphical view to express the topic words and their relations, words with higher probability reflect some interesting aspects of the topic, and word pairs (*<t, s>*) with higher probability reflect their correlations, so the topic network consists of these two parts. Algorithm 1 shows how to construct a text network for a given topic.

Algorithm 1:
Input: topic id *k*
Output: a topic network *<V, E>* of *k*, where *V* and *E* are vertices and edges.

Table 1: WordTopicProb, it stores the conditional probability $p(t|k)$

Field	Type	Example	Description
word	varchar(100)	fuel	word
topicID	int	19	topic ID
CWK	int	745	$n_k^{(t)}$
CK	int	3673	$\sum_t n_k^{(t)}$
PWK	float	0.2028	ϕ_k^t

Table 2: WordPairTopicProb, it stores the conditional probability $p(t, s|k)$

Field	Type	Example	Description
wordA	varchar(100)	tank	word A
wordB	varchar(100)	fuel	word B
topicID	int	19	topic ID
CK	int	3673	$\sum_t n_k^{(t)}$
CWAK	int	186	$n_k^{(t)}$
CWABK	int	143	$n_{k,t}^{(s)}$
PA_K	float	0.0506	$p(t|k)$
PB_AK	float	0.7688	$p(s|k, t)$
PAB_K	float	0.0389	$p(t, s|k)$

1. Initialize V and E to empty set
2. *select top m word from WordTopicProb where topicid = k order by pwk desc*, denote it as set A
3. *select top n worda, wordb from WordpairTopicProb where topicid = k order by pab_k desc*, denote it as set B
4. for each *word* in A, put it into V
5. for each *worda* in B, put it into V if not duplicate
6. for each *wordb* in B, put it into V if not duplicate
7. for each <*worda*, *wordb*> in B, put it into E
8. return <V, E>

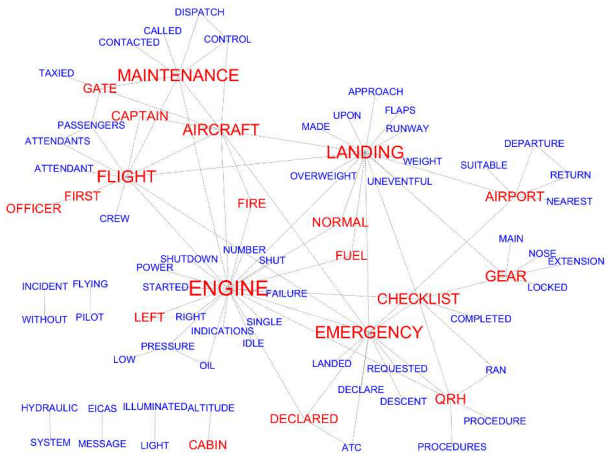


Figure 4: a topic network about "Aircraft Equipment Problem Critical" of year 2013 for all airports

A topic network about *"Aircraft Equipment Problem Critical"* of year 2013 for all airports is shown in figure 4. In the network, the red color words are topic words in set A, and their fonts are in ratio with their probabilities. The other words are blue. We use smaller fonts to distinguish them from the red ones. The topic network well exhibits the topic words distribution and their relations, and word pairs also provide interesting information, such as *engine shutdown, engine failure, overweight landing* etc. Some words are not connected to the main component, but they also provide some useful information, such as *hydraulic system, EICAS (Engine Indication and Crew Alerting System) message* etc. In addition, there are some clusters in figure 4, such as *flight / aircraft / maintenance* about aircraft maintenance, and *enginee / emergency / landing / airport* about emergency landing.

4. OLAP on topic network cube

Unlike an ordinary data cube, the data in topic network cube could not be aggregated directly, then how to do multi-scale analysis on it? What would the topic network be when query conditions are changed? All the dimensions in topic network cube can be divided into two categories: the first is ordinary dimensions such as time and location, and the second is topic dimensions as shown in figure 3.

4.1. Multi-scale analysis on ordinary dimensions

When query conditions about ordinary dimensions are changed, the involved documents are also changed. For example, suppose we are analyzing all documents of year 2013 currently, and next we want to *drill-down* through time dimension and only analyze the 4th quarter of this year, then only the documents reported in the 4th quarter should be considered. Because topic words stored in table *WordTopicProb* and table *WordPairTopicProb* are about the whole year, we need to recalculate its topic words distribution. To do this, table *SearchDocument* should be refilled by the following SQL statements:

truncate table searchdocuments
insert into searchdocuments select documents. from documents inner join dates on documents.strdate = dates.strdate where dates.intyear = 2013 and dates.intquarter=4*

After refilling, the other two tables *WordTopicProb* and *WordPairTopicProb* should also be refilled using the method discussed in section 3.1, with only the difference that adding *"where documentid in (select documentid from searchdocuments)"* in its where clause. For convenience, we create two procedures to refill these two tables (see Appendix A.1-3). Note that content of tables in the pink part does not change, which means that Labeled LDA never needs to be recalculated again.

4.2. Multi-scale analysis on topic dimensions

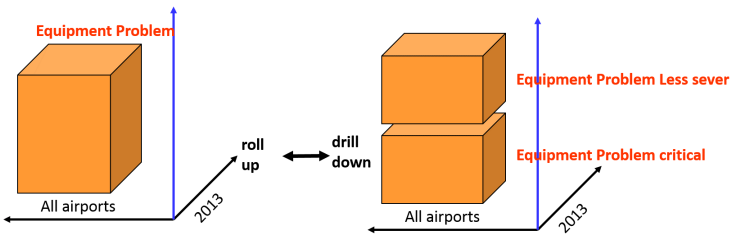


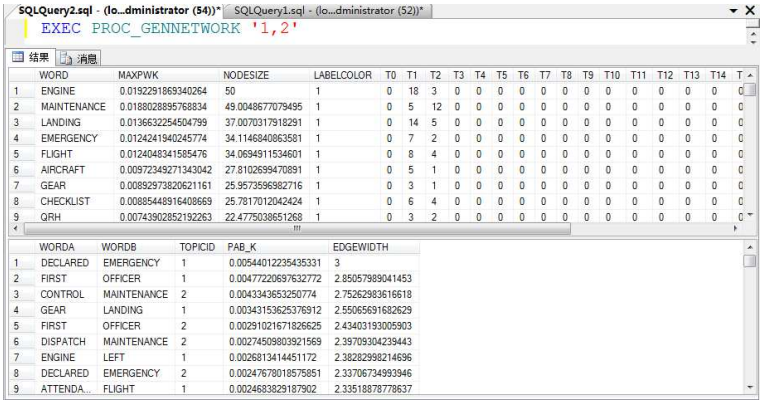
Figure 5: roll up or drill down through topic dimension

When rolling up or drilling down through topic dimensions, since topic words stored in the light blue part tables are about all topics, their contents are still usable, and we only need to search topic words from them. For example, suppose we are now analyzing the topic *"Equipment problem critical"*, and next we want to analyze a higher level topic of *"Equipment problem"*, what we need to do is only using *"topicid in (1,2)"* instead of *"topicid = 1"* in algorithm 1, where 1 is the topicid of *"Equipment problem critical"*, and 2 is the topicid of *"Equipment problem less sever"*.

For convenience, we create a procedure *proc_gennetwork* as listed in appendix A.4. For example, if we want to get the topic network about *"Equipment problem"* in year 2013, we only need to run the procedure *exec proc_gennetwork '1,2'*, and the outputs are shown in figure 6. The upper part is the node table, and the lower part is the edge table. To make a better effect in visualization, we set the font size of nodes and the width of edges according to their probabilities. In the node table, T0 to T22 indicates which topics a word may belong to, such as word *"maintenance"* belongs to topic 1 and 2. Its corresponding number of 5 and 12 means that it has 5 connections to topic 1 and 12 connections to topic 2. With these 2 numbers, we can use a pie chart in a visualization software to express its overlapping feature, as shown in figure 7.

Figure 7 is the topic network of *"Equipment problem"* in year 2013 for all airports, in which the blue lines are about *"Equipment Problem Critical"*, and the red lines are about *"Equipment Problem Less Sever"*. There

may be more than one connections between two words, such as *emergency* and *declared*. One word may also belong to more than one topics, such as *maintenance* belongs to topic 1 and topic 2. By line colors, figure 7 can be separated into two parts, where *maintenance* mainly connects to *"Equipment Problem Less Sever"*, while *engine* mainly connects to *"Equipment Problem Critical"*. All topic words have good explanations, and the topic network also well exhibits the topic distributions.



	WORD	MAXPWK	NODESIZE	LABELCOLOR	T0	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13	T14	T
1	ENGINE	0.0192291869340264	50	1	0	18	3	0	0	0	0	0	0	0	0	0	0	0	0	0
2	MAINTENANCE	0.0188028895768834	49.0048677079495	1	0	5	12	0	0	0	0	0	0	0	0	0	0	0	0	0
3	LANDING	0.0136632254504799	37.0070317918291	1	0	14	5	0	0	0	0	0	0	0	0	0	0	0	0	0
4	EMERGENCY	0.0124241940245774	34.1146840863581	1	0	7	2	0	0	0	0	0	0	0	0	0	0	0	0	0
5	FLIGHT	0.0124048341585476	34.0694911534601	1	0	8	4	0	0	0	0	0	0	0	0	0	0	0	0	0
6	AIRCRAFT	0.00972349271343042	27.8102699470891	1	0	5	1	0	0	0	0	0	0	0	0	0	0	0	0	0
7	GEAR	0.00892973820621161	25.9573596982716	1	0	3	1	0	0	0	0	0	0	0	0	0	0	0	0	0
8	CHECKLIST	0.00885448916408669	25.7817012042424	1	0	6	4	0	0	0	0	0	0	0	0	0	0	0	0	0
9	QRH	0.00743902852192263	22.4775038651268	1	0	3	2	0	0	0	0	0	0	0	0	0	0	0	0	0

	WORDA	WORDB	TOPICID	PAB_K	EDGEWIDTH
1	DECLARED	EMERGENCY	1	0.00544012235435331	3
2	FIRST	OFFICER	1	0.00477220697632772	2.85057989041453
3	CONTROL	MAINTENANCE	2	0.0043343653250774	2.75262983616618
4	GEAR	LANDING	1	0.00343153625376912	2.55065691682629
5	FIRST	OFFICER	2	0.00291021671826625	2.43403193005903
6	DISPATCH	MAINTENANCE	2	0.00274509803921569	2.39709304239443
7	ENGINE	LEFT	1	0.0026813414451172	2.38282998214696
8	DECLARED	EMERGENCY	2	0.00247678018575851	2.33706734993946
9	ATTENDA...	FLIGHT	1	0.0024683829187902	2.33518878778637

Figure 6: the searching result of "Equipment problem" in year 2013 for all airports, in which the upper part is the node table, and the following part is the edge table

5. Experiment

In this section, we introduce our experiment of OLAP analysis using topic network cube on ASRS reports. ASRS (Aviation Safety Reporting System, http://asrs.arc.nasa.gov) was established to collect incident reports voluntarily submitted by pilots, air traffic controllers, dispatchers, flight attendants, maintenance technicians and other related parties. It is hoped that the information from the ASRS system can support the aviation authorities to identify and address problems in the National Aviation System. We searched 22 abnormal events in year 2013, and got 4279 ASRS reports. In the preprocessing step, some words including symbols, numbers, words with only one character and stop words (http://code.google.com/p/stop-words/) are dropped. The final vocabulary has 19,324 distinct words.

At first, some initial steps must be done before our multi-scale analysis:

1. Do Labeled LDA analysis for all the documents
2. Save its Gibbs sampling outputs into table Words
3. Fill table WordABK as discussed in section 3.1

- **Case 1: equipment problems analysis for all reports in year 2013**

  Firstly, we query all the involved documents into table *SearchDocuments* using:
  truncate table searchdocuments
  *insert into searchdocuments select * from documents*
  And then, we refill table *WordTopicProb* and *WordPairTopicProb* using:
  exec proc_fullwordtopicprob
  exec proc_fullwordpairtopicprob
  And now, we can get network about topic *"equipment problem critical"* of year 2013 for all airports using: *exec proc_gennetwork '1'*

  The corresponding topic network is shown in figure 4. If we want to see the network of topic *"equipment problem"*, we can simply use *exec proc_gennetwork '1,2'* as discussed in section 4.2, and the result is shown in figure 7.

- **Case 2: Flight deck event analysis for all airports in year 2013**

  Flight Deck Event has two sub topics, the first is *"Flight Deck / Cabin / Aircraft Event Other / Unknown"* with topicid 15, and the second is *"Flight Deck / Cabin / Aircraft Event Smoke / Fire / Fumes / Odor"* with topicid 16. Because the query conditions on ordinary dimensions are same as in case 1, we just run a simple SQL: *exec proc_gennnetwork '15,16'* in order to get the topic network of *Flight Deck Event*, and the result is shown in figure 8. We can clearly see the difference between these two topics through this figure. words with red lines are about *"Flight Deck / Cabin / Aircraft Event Smoke / Fire / Fumes / Odor"*, which is mainly about smoke, fumes or fire in cabin or cockpit; while words with blue lines are about *"other topics"*, which is mainly about cart or carts event in galley or aisle.

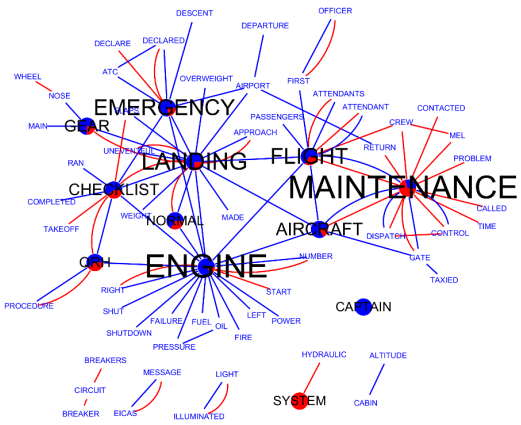


Figure 7: topic network of *"Equipment problem"* in year 2013

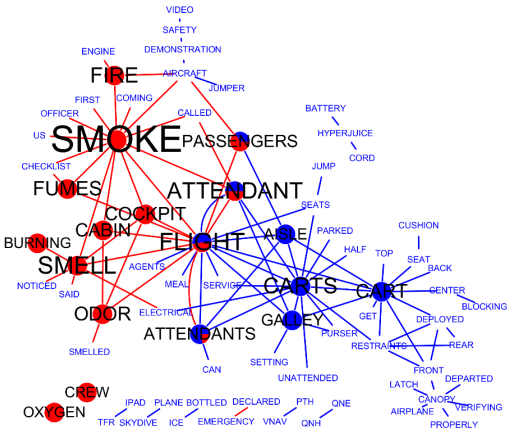


Figure 8: topic network of *"Flight Deck Event"* in year 2013

- **Case 3: weather analysis in different seasons**

  To get network of topic *"Inflight Event / Encounter Weather / Turbulence"* in summer, we use the following SQL statements, and the result is shown in figure 9 (left part).

  truncate table searchdocuments
  insert into searchdocuments select documents. from documents inner join dates on*
  documents.strdate=dates.strdate where dates.intyear=2013 and dates.intmonth in (6,7,8)
  exec proc_fullwordtopicprob
  exec proc_fullwordpairtopicprob
  exec proc_gennetwork '21'

  To get network about topic "Inflight Event / Encounter Weather / Turbulence" in winter, we use the following SQL statements, and the result is shown in figure 9 (right part).

  truncate table searchdocuments
  insert into searchdocuments select documents. from documents inner join dates on*
  documents.strdate=dates.strdate where dates.intyear=2013 and dates.intmonth in (12,1,2)
  exec proc_fullwordtopicprob
  exec proc_fullwordpairtopicprob
  exec proc_gennetwork '21'

Figure 9 is the topic network of abnormal weather in summer (left) and winter (right). Many topic words of them are same, such as *weather, encountered moderate* or *severe turbulence*. At the same time, there are also some different topic words, for example, *windshear, microburst* and *thunderstorms* in summer, while *ice, icing* and *heat* in winter.

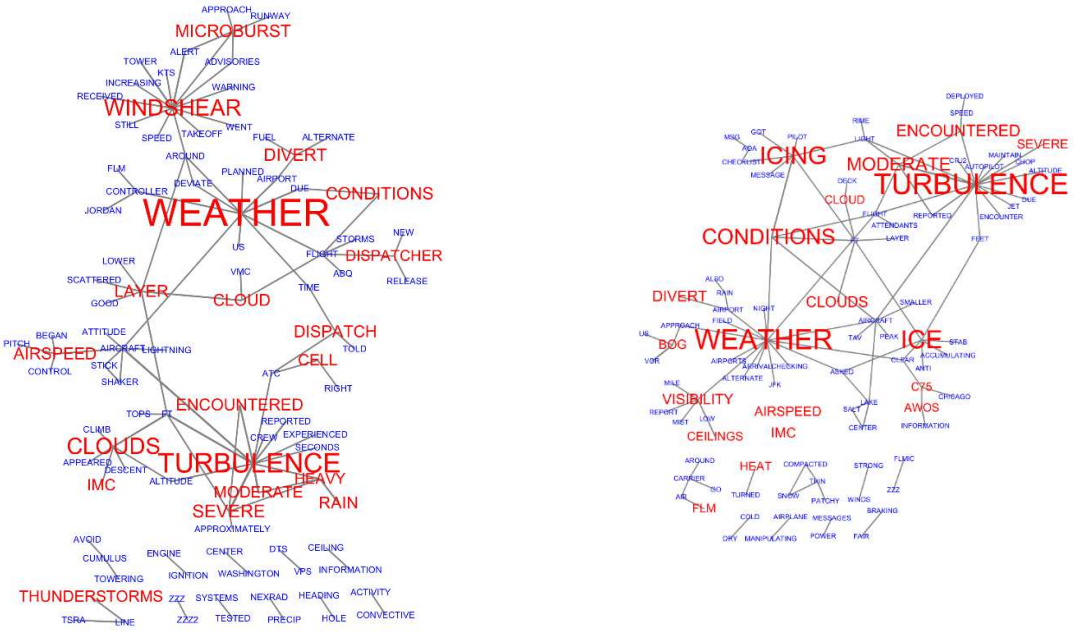


Figure 9: topic network of *"Inflight Event / Encounter Weather / Turbulence"* in different seasons. Left part: summer (Jun.-Aug.). Right part: winter (Dec.-Feb.)

6. Conclusion

This paper introduces a novel concept of topic network cube and its applications on OLAP analysis of multidimensional text databases. A topic network connects related topic words, not only giving a direct distribution of topic words for specific topics, but also expressing the complex relations among these topic words. Clusters of a topic network further reveals subtopics of it, providing rich and useful information for analyzers. A fact constellation storage model of topic network cube is also discussed. This model contains several dimensions including ordinary dimensions and a topic dimension. Together with GL-LDA and procedures we created, this model can efficiently generate a topic network by a few simple SQL statements, making it an easy way to do multi-scale analysis on every dimension in different granularity.

References

[1] Chaudhuri S, Dayal U, An overview of data warehousing and OLAP technology, SIGMOD, Tucson, Arizona, USA, 26 (1997): 65-74.
[2] Lin C X, Ding B, Han J, et al. Text Cube: Computing IR Measures for Multidimensional Text Database Analysis[C]// 8th IEEE International Conference on Data Mining. IEEE, 2008:905-910.
[3] Zhang D, Zhai C X, Han J, et al. Topic modeling for OLAP on multidimensional text databases: topic cube and its applications[J]. Statistical Analysis & Data Mining, 2009, 2(5):378-395.
[4] Chen C, Yan X, Zhu F, et al. Graph OLAP: a multi-dimensional framework for graph data analysis[J]. Knowledge & Information Systems, 2009, 21(1):41-63.
[5] Loudcher S, Jakawat W, Morales E P S, et al. Combining OLAP and Information Networks for Bibliographic Data Analysis: A Survey[J]. Scientometrics, 2015, 103(2):471-487.
[6] Bimonte S, Tchounikine A, Miquel M. GeoCube, a Multidimensional Model and Navigation Operators Handling Complex Measures: Application in Spatial OLAP[M]// Advances in Information Systems. Springer Berlin Heidelberg, 2006:100-109.
[7] Hofmann T. Unsupervised learning by probabilistic latent semantic analysis[J]. Machine learning, 2001, 42(1-2): 177-196.
[8] Blei D M, Ng A Y, Jordan M I. Latent dirichlet allocation[J]. Journal of machine learning research, 2003, 3: 993-1022.
[9] Ramage D, Hall D, Nallapati R, et al. Labeled LDA: A supervised topic model for credit attribution in multi-labeled corpora[C]//Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, EMNLP. ACL, 2009: 248-256.
[10] Blei D M, Mcauliffe J D. Supervised Topic Models[C]// Advances in Neural Information Processing Systems 2007, MIT Press, 2007:121-128.

[11] Church K W, Hanks P. Word association norms, mutual information, and lexicography[J]. Computational linguistics, 1990, 16(1): 22-29.
[12] Turney P. Mining the web for synonyms: PMI-IR versus LSA on TOEFL[C]//Proceedings of the 12th European Conference on Machine Learning, ECML. 2001: 491-502.
[13] Smith A, Chuang J, Hu Y, et al. Concurrent Visualization of Relationships between Words and Topics in Topic Models[C]// The Workshop on Interactive Language Learning, Visualization, and Interfaces. 2014.
[14] Han L, Finin T, Mcnamee P, et al. Improving Word Similarity by Augmenting PMI with Estimates of Word Polysemy[J]. IEEE Transactions on Knowledge & Data Engineering, 2013, 25(6):1307-1322.
[15] Zhang Z Y, Huo W G. A Topic Text Network Construction Method Based on PL-LDA Model[J]. Complex Systems and Complexity Science, 2017, 14(1):52-57 (in Chinese)
[16] Zhang Z Y, Yang H J, Zhao Y. Topical Text Network Construction Method Based on Gibbs Sampling Results[J]. Computer Engineering, 2017, 43(6):150-157 (in Chinese)

Appendix

A.1 SQL statements of procedure PROC_FULLWORDTOPICPROB (β_t was set to 0.5 for convenience)

```
CREATE PROCEDURE PROC_FULLWORDTOPICPROB AS
BEGIN
TRUNCATE TABLE WORDTOPICPROB
INSERT INTO WORDTOPICPROB (WORD, TOPICID, CWK)
SELECT WORD, TOPICID, COUNT(*)+0.5 FROM WORDS WHERE DOCUMENTID IN (SELECT
DOCUMENTID FROM SEARCHDOCUMENTS) GROUP BY WORD, TOPICID
DECLARE @TOPICID INT
DECLARE @TOTAL FLOAT
DECLARE CUR CURSOR FOR SELECT TOPICID FROM TOPICS
OPEN CUR
FETCH CUR INTO @TOPICID
WHILE @@FETCH_STATUS=0
BEGIN
SELECT @TOTAL=SUM(CWK) FROM WORDTOPICPROB WHERE TOPICID=@TOPICID
UPDATE WORDTOPICPROB SET CK=@TOTAL, PWK=CWK/@TOTAL WHERE TOPI-
CID=@TOPICID
FETCH CUR INTO @TOPICID
END
CLOSE CUR
DEALLOCATE CUR
END
```

A.2 SQL statements of procedure PROC_FULLWORDPAIRTOPICPROB (β_t was set to 0.5 for convenience)

```
CREATE PROCEDURE PROC_FULLWORDPAIRTOPICPROB AS
BEGIN
CREATE TABLE #TMP(
WORDA VARCHAR(100) NOT NULL,
WORDB VARCHAR(100) NOT NULL,
TOPICID INT NOT NULL,
CK FLOAT,
CWAK FLOAT,
CWABK FLOAT,
PA_K FLOAT,
PB_AK FLOAT,
PAB_K FLOAT)
INSERT INTO #TMP
(WORDA, WORDB, TOPICID, CWABK)
SELECT WORDA, WORDB, WORDATOPICID, COUNT(*)+0.5 FROM WORDABK
WHERE DOCUMENTID IN (SELECT DOCUMENTID FROM SEARCHDOCUMENTS)
GROUP BY WORDA, WORDB, WORDATOPICID
UPDATE #TMP
SET CWAK=WORDTOPICPROB.CWK, CK=WORDTOPICPROB.CK, PA_K=WORDTOPICPROB.PWK
FROM #TMP INNER JOIN WORDTOPICPROB ON #TMP.WORDA=WORDTOPICPROB.WORD
AND #TMP.TOPICID=WORDTOPICPROB.TOPICID
UPDATE #TMP
SET PB_AK=CWABK / CWAK, PAB_K=PA_K*CWABK/CWAK
TRUNCATE TABLE WORDPAIRTOPICPROB
INSERT INTO WORDPAIRTOPICPROB(WORDA,WORDB,TOPICID,PAB_K)
SELECT DBO.FUN_WORDABORDER(WORDA,WORDB),
DBO.FUN_WORDABORDER(WORDA,WORDB),TOPICID,AVG(PAB_K)
FROM #TMP GROUP BY DBO.FUN_WORDABORDER(WORDA,WORDB), TOPICID
UPDATE WORDPAIRTOPICPROB
SET WORDA=LEFT(WORDA,CHARINDEX(',',WORDA)-1),WORDB=RIGHT(WORDB,LEN(WORDB)-
CHARINDEX(',',WORDB))
END
```

A.3 SQL statements of function FUN_WORDABORDER

If word pair <A, B>and <B, A>of one topic both exist, mean value of them is taken to omit the effect of word pair orders. A user defined function FUN_WORDABORDER is then created to fulfill this job.

```
CREATE FUNCTION DBO.FUN_WORDABORDER(
@WORDA VARCHAR(100),@WORDB VARCHAR(100)) RETURNS VARCHAR(200) AS
BEGIN
DECLARE @X VARCHAR(200)
IF @WORDA <@WORDB
SET @X=@WORDA+','+@WORDB
ELSE
SET @X=@WORDB+','+@WORDA
RETURN @X
END
```

A.4 SQL statements of procedure PROC_GENNETWORK

```
CREATE PROCEDURE PROC_GENNETWORK(@TOPICLIST VARCHAR(200)) AS
BEGIN
CREATE TABLE #TMPNODE(
WORD VARCHAR(100),
TOPICID INT,
PWK FLOAT)
CREATE TABLE #TMPEDGE(
WORDA VARCHAR(100),
WORDB VARCHAR(100),
TOPICID INT,
PAB_K FLOAT,
EDGEWIDTH FLOAT)
DECLARE @SQL VARCHAR(8000)
SET @SQL=''
SET @SQL=@SQL+'INSERT INTO #TMPNODE'+CHAR(10)
SET @SQL=@SQL+'SELECT TOP 20 WORD, TOPICID, PWK FROM WORDTOPICPROB'+CHAR(10)
SET @SQL=@SQL+'WHERE TOPICID IN ('+@TOPICLIST+')' + CHAR(10)
SET @SQL=@SQL+'ORDER BY PWK DESC'
EXECUTE(@SQL)
SET @SQL=''
SET @SQL=@SQL+'INSERT INTO #TMPEDGE' + CHAR(10)
SET @SQL=@SQL+'SELECT TOP 100 WORDA, WORDB, TOPICID, PAB_K,0 FROM
WORDPAIRTOPICPROB' + CHAR(10)
SET @SQL=@SQL+'WHERE TOPICID IN (' + @TOPICLIST + ')' + CHAR(10)
SET @SQL=@SQL+'ORDER BY PAB_K DESC'
EXECUTE(@SQL)
--add solo nodes
INSERT INTO #TMPEDGE(WORDA,TOPICID)
SELECT WORD, TOPICID FROM #TMPNODE
WHERE NOT EXISTS(
SELECT * FROM #TMPEDGE WHERE (#TMPEDGE.WORDA=#TMPNODE.WORD
AND #TMPEDGE.TOPICID=#TMPNODE.TOPICID) OR
(#TMPEDGE.WORDB=#TMPNODE.WORD AND #TMPEDGE.TOPICID=#TMPNODE.TOPICID) )
INSERT INTO #TMPNODE(WORD)
SELECT DISTINCT WORDA FROM #TMPEDGE
WHERE WORDA NOT IN (SELECT WORD FROM #TMPNODE)
INSERT INTO #TMPNODE(WORD)
SELECT DISTINCT WORDB FROM #TMPEDGE
WHERE WORDB NOT IN (SELECT WORD FROM #TMPNODE)
CREATE TABLE #NODETABLE(
WORD VARCHAR(100),
MAXPWK FLOAT,
NODESIZE FLOAT,
LABELCOLOR INT,
T0 INT, T1 INT, T2 INT, ..., T22 INT)
INSERT INTO #NODETABLE
SELECT A.WORD,
MAXPWK=MAX(A.PWK),
NODESIZE=0,
LABELCOLOR=1,
T0=SUM(CASE B.TOPICID WHEN 0 THEN 1 ELSE 0 END),
T1=SUM(CASE B.TOPICID WHEN 1 THEN 1 ELSE 0 END),
.....
T22 = SUM(CASE B.TOPICID WHEN 22 THEN 1 ELSE 0 END)
FROM (SELECT WORD,PWK=MAX(PWK) FROM #TMPNODE GROUP BY WORD) AS
A INNER JOIN #TMPEDGE B ON (
A.WORD=B.WORDA OR A.WORD=B.WORDB)
GROUP BY A.WORD
DECLARE @X FLOAT
DECLARE @Y FLOAT
SELECT @X = MAX(MAXPWK),@Y=MIN(MAXPWK) FROM #NODETABLE
UPDATE #NODETABLE
SET NODESIZE = (MAXPWK-@Y)/(@X-@Y) * 30.0 + 20.0
UPDATE #NODETABLE
SET LABELCOLOR = 0,
T0=0,T1=0,T2=0,T3=0,...,T22=0
WHERE MAXPWK IS NULL
UPDATE #NODETABLE
SET NODESIZE=12 WHERE NODESIZE IS NULL
SELECT @X = MAX(PAB_K),@Y=MIN(PAB_K) FROM #TMPEDGE
UPDATE #TMPEDGE
SET EDGEWIDTH = (PAB_K-@Y)/(@X-@Y) * 1.0 + 2.0
SELECT * FROM #NODETABLE ORDER BY NODESIZE DESC
SELECT * FROM #TMPEDGE ORDER BY EDGEWIDTH DESC
END
```