

Multi-Dimensional, Phrase-Based Summarization in Text Cubes

Fangbo Tao[†], Honglei Zhuang[†], Chi Wang Yu[‡], Qi Wang[†], Taylor Cassidy[§],
Lance Kaplan[§], Clare Voss[§], Jiawei Han[†]

[†] *Department of Computer Science, UIUC*

[‡] *Microsoft Research*

[§] *US Army Research Laboratory*

Abstract

To systematically analyze large numbers of textual documents, it is often desirable to manage documents (and their metadata) in a multi-dimensional text database (Text Cube). Such structure provides flexibility of understanding local information with different granularities. Moreover, the contextualized analysis derived from cube structure often yields comparative insights. To quickly digest the content of subsets of documents in the multi-dimensional context, we study the problem of phrase-based summarization of a subset of documents of interest. We propose a new phrase ranking measure to leverage the relation between document subsets induced by multi-dimensional context and identify phrases that truly distinguish the queried subset of documents from neighboring subsets (i.e., background). Our quality evaluation suggests the new measure involving dynamic, query-dependent background generation is more effective than previous measures using the whole corpus as a static background for finding representative phrases. Computing this measure is more expensive due to the need of access to many subsets of documents to answer one query. We develop a cube-based analytical platform that implements an efficient solution by materializing a deliberately selected part of statistics, and using these statistics to perform online query processing within a constant latency constraint. Our experiments in a large news dataset demonstrate the efficiency in both query processing time and storage cost.

1 Introduction

With ever more massive datasets accumulating in text repositories (e.g., news articles, business reports, customer reviews, etc.), it is highly desirable to conduct multi-dimensional analysis on text data, where the dimensions correspond to multiple meta attributes (e.g., category, date/time, location, author, etc.) associated with the documents. The dimensions provide rich context to partition the documents and relate them, and users can use these dimensions to navigate to a subset of documents of interest from a huge corpus. Typically, structured/relational data has been handled by relational database systems, and such systems also provide some text indexing and search capabilities to assist text data stored in such (extended) relational database systems. However, such kind of systems often suffer from the following limitations.

- It can hardly support systematic analysis of large collections of free text in multi-dimensional way, although such text data is ubiquitous in real-world;

Copyright 2016 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

Bulletin of the IEEE Computer Society Technical Committee on Data Engineering

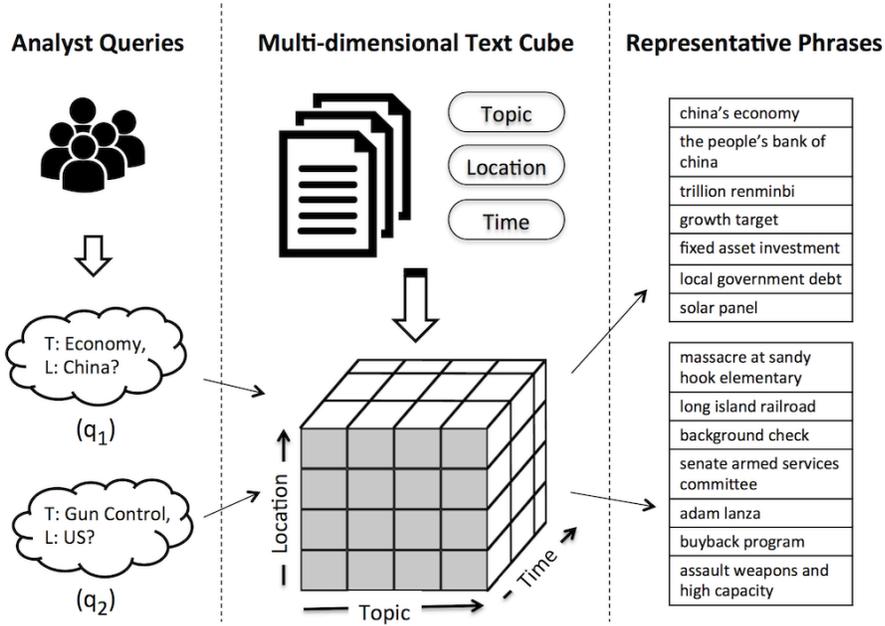


Figure 1: Illustration of phrase-based summarization in text cubes

- It usually does not support data cube technologies on text data and multidimensional text mining, although it is obvious that text mining and data cube technologies can mutually enhance each other; and
- There is a lack of a general platform that can support integrated multi-dimensional analysis of structured and text data, on top of which many powerful analysis methods and tools can be developed, experimented and refined, such as viewing such data sets as interconnected information networks and further applying information network analysis technology.

In this paper, we propose a multi-dimensional perspective of large-scale text corpora. In particular, we introduce the framework of *Text Cube* [8] and its analytical platform [14]. To help users efficiently explore the text cubes, we study the problem of *phrase-based summarization in multi-dimensional context*: given user-specified dimensions and their values, return top- k phrases that characterize the corresponding set of documents. The resulting phrases carry rich semantics and may benefit various downstream applications, e.g., *text summarization*.

Example 1. Suppose a multi-dimensional text database is constructed from *New York Times* news repository with three meta attributes: *Location*, *Topic*, and *Time*, as shown in Figure 1. An analyst may pose multi-dimensional queries such as: (q_1) : $\langle \text{China, Economy} \rangle$ and (q_2) : $\langle \text{US, Gun Control} \rangle$. Each query asks for summary of a cell defined by two dimensions *Location* and *Topic*. What kind of cell summary does she like to see? Frequent unigrams such as *debt* or *senate* are not as informative as multi-word phrases, such as *local government debt* and *senate armed service committee*. The phrases preserve better semantics as integral units rather than as separate words.

Generally, three criteria should be considered when ranking representative phrases in a selected multi-dimensional cell: (i) integrity: a phrase that provides integral semantic unit should be preferable over non-integral unigrams, (ii) popularity: popular in the selected cell (i.e., selected subset of documents), and (iii) distinctiveness: distinguish the selected cell from other cells.

The remainder of the paper proceeds as follows. Section 2 introduces the framework of *Text Cube* and explains the power of converting text corpora to text cubes. The phrase-based summarization is proposed within the framework. Its effectiveness is evaluated by various experiments. Section 3 introduces the computational

platform for multi-dimensional text analysis, including the computational optimization for phrase-based summarization. Section 4 concludes the paper.

2 Multi-dimensional Text Analysis

In this section, we formally define the concept of *Text Cube*, the *Phrase-based Summarization* problem, the three phrase ranking criteria, and multiple experimental results to elaborate the effectiveness of phrase summarization.

Several pieces of related work have been proposed along this research line. Text Cube [8] takes a multi-dimensional view of textual collections and proposed OLAP-style *tf* and *idf* measures. Besides that, [7, 11] also proposed OLAP-style measures on term level using only local frequency, which cannot serve as effective semantic representations. [16, 4] focused on interactive exploration framework in text cubes given keyword queries, without considering the semantics in raw text. Similarly, R-Cube [10] is proposed where user specify an analysis portion by supplying some keywords and a set of cells are extracted based on relevance. Another related topic is Faceted Search [6, 15, 2, 3], which dynamically aggregates information for an ad-hoc set of documents. the aggregation is usually conducted on meta data (called *facets*), not document content.

2.1 Text Cube

Similar to traditional multi-dimensional data cubes, a *text cube* [8] is a data model but over text collection *DOC* that has metadata for documents. The metadata can be either extrinsic attributes of the documents, such as classification taxonomy, or intrinsic information extracted from the documents, such as named entities mentioned in them. In this paper, we focus on single-valued categorical metadata, and leave other types of metadata to future work. We assume there are n categorical attributes (*i.e.*, *dimensions*) associated with each document in *DOC*. For example, a news article in *NYT* corpus is represented as (*Jan 2012, China, Economy, 'After a sharp economic slowdown through much of last year...'*). It denotes that the 'Time' of the article is *Jan 2012*, 'Location' is *China* and 'Topic' is *Economy*.

The dimensions provide valuable context for each document. Like a traditional data cube, all distinct values of one dimension are organized in a *dimension hierarchy*. For i -th dimension, the dimension hierarchy \mathcal{A}_i is a tree where the root is denoted as '*'. Each non-root node is a value in that dimension. The parent node of a dimension value a_i is denoted as $par(a_i)$, and the set of direct descendants of a_i is denoted as $des(a_i)$. For example, Figure 3 illustrates a partial dimension hierarchy about 'topics' in *NYT* corpus. It is a tree of height 4, with a root node '*'. $par(Gun\ Control) = Domestic\ Issues$ and $des('*') = \{Economy, Sports, Politics\}$.

Formally, we have the following definition.

Definition 1 (Multi-dimensional Text Cube): A

text cube is defined as $\mathcal{TC} = (\mathcal{A}_1, \mathcal{A}_2, \dots, \mathcal{A}_n, \mathcal{DOC})$, where \mathcal{A}_i is a dimension hierarchy. Each document is in the form of $(a_1, a_2, \dots, a_n, d)$, where $a_i \in \mathcal{A}_i \setminus \{*\}$ is a dimension value for \mathcal{A}_i and d is a string of the content. A cell c in the cube is represented as $(a_1, \dots, a_n, \mathcal{D}_c)$, where $a_i \in \mathcal{A}_i$, and $\mathcal{D}_c \subseteq \mathcal{DOC}$ is the subset of documents contained in cell c . For notation simplicity, we use $\langle a_{t_1}, \dots, a_{t_k} \rangle$ to refer to a cell with non-* dimension values $\{a_{t_1}, \dots, a_{t_k}\}$.

Example 3. Figure 2 illustrates a mini example of news article text cube, with 3 dimensions (Time, Location and Topic) and 9 documents d_1-d_9 . The Time dimension is derived from extrinsic attribute but Location and Topic are extracted by information extraction as in [13]. We pick 7 non-empty cells, where the top four are leaf cells without '*' dimensions, *e.g.*, (*Jan 2012, China, Economy, \{d_1, d_2\}*). The root cell (entire corpus) is represented as $(*, *, *, \{d_1-d_9\})$.

Text cube provides a framework for organizing text documents using meta-information. In particular, the cell space defined above embeds the inter-connection between different subsets of text. To capture those semantically close cells, we define *context* of a cell c as a composition of three parts.

Dimensions			Text Data
Year	Location	Topic	\mathcal{DOC}
2011	China	Economy	$\{d_1, d_2\}$
2012	China	Economy	$\{d_3, d_4, d_5\}$
2012	US	Gun Control	$\{d_6, d_7\}$
2013	US	Economy	$\{d_8, d_9\}$
*	China	Economy	$\{d_1, \dots, d_5\}$
2012	*	*	$\{d_3, \dots, d_7\}$
*	*	*	$\{d_1, \dots, d_9\}$

Figure 2: Mini Example of *NYT* Corpus

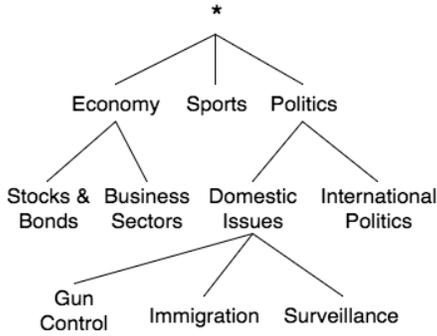


Figure 3: Hierarchy of *Topic*

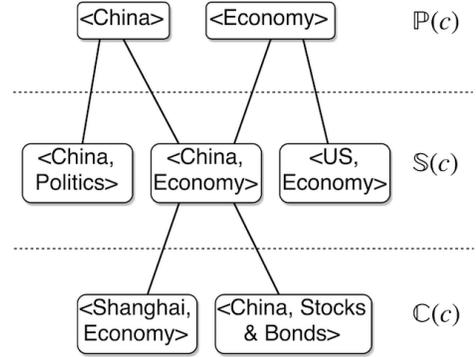


Figure 4: *Context* of cell $\langle \text{China, Economy} \rangle$

Definition 2 (Cell Context): The context of cell $c = \langle a_{t_1}, \dots, a_{t_k} \rangle$ is defined as $\mathbb{P}(c) \cup \mathbb{S}(c) \cup \mathbb{C}(c)$, where:

- Parent set is defined as $\mathbb{P}(c) = \{ \langle a_{t_1}, \dots, \text{par}(a_i), \dots, a_{t_k} \rangle \mid i \in t_1, \dots, t_k \}$. Each parent cell is found by changing exactly one non-* dimension value in cell c into its parent value;
- Children set is defined as $\mathbb{C}(c) = \{ c' \mid c \in \mathbb{P}(c') \}$. Each child cell is found by either changing one * value into non-* or by replacing it by one of the child values; and
- Sibling set is defined as $\mathbb{S}(c) = \{ c' \mid \mathbb{P}(c) \cap \mathbb{P}(c') \neq \emptyset \}$. Each sibling cell must share one parent with cell c .

Example 4. Figure 4 illustrates the partial context of cell $c = \langle \text{China, Economy} \rangle$. The parent set $\mathbb{P}(c)$ contains $\langle \text{China} \rangle$ and $\langle \text{Economy} \rangle$, sibling set $\mathbb{S}(c)$ has $\langle \text{China, Politics} \rangle$ and $\langle \text{US, Economy} \rangle$ and children $\mathbb{C}(c)$ contains $\langle \text{Shanghai, Economy} \rangle$ and $\langle \text{China, Stocks \& Bonds} \rangle$.

2.2 Cube Perspective of Text Corpora

Organizing a text corpus into a text cube provides various possibilities to substantially improve user experience in browsing, retrieving, and analyzing large scale textual data.

- **Enriched horizon.** Multi-dimensional structure grants analysts with an enriched mine of knowledge to be discovered. For example, without a multi-dimensional structure, an analyst can either perform statistics on the entire corpus, or simply perform statistics on a single documents. However, when the corpus is organized as

a text cube, the analyst is able to study the connection between various statistics of documents and different categories. For example, one may check whether there is a correlation between the frequency of different words and the *publishing time* in a news corpus, to understand how the usage of a word varies. Similarly, one can also examine the key phrases of a certain category of documents (e.g. news articles about “Brazil”) to obtain a better picture of the subset of interest. The additional meta-information not only allows analysts to deepen their understanding on different facets of the document data sets, but also provides them with better insights of the dimensions per se.

- **Contextualized analysis.** Multi-dimensional structure also enables the analysts to conduct analysis with a certain context. A analyst may be interested on a specific subset of documents, for example, news articles about “China Economy”. However, documents from other relevant subsets may also be useful in better understanding this concept, like “Japan Economy” or “US Economy”. Generally, they help analysts in comparative studies, to better understand the features the document subset shares with other subsets, and the features unique to the subset. As a more specific example, suppose an analyst is interested in summarizing key phrases of “China Economy”, it is helpful to remove phrases overlapping with “Japan Economy” or “US Economy”, such as “banking” or “currency”, as they do not distinguish the subset of interest from others.

2.3 Phrase-based Summarization

This paper deals with the problem of mining representative phrases to serve as summary, in particular within multi-dimensional text cubes. A phrase is a multi-word sequence served as an integral semantic unit. The representative phrases for a cell, are the phrases that characterize the semantics of the selected documents. There is no universally accepted standard of being *representative*. Here we operationalize a definition in terms of three criteria.

- **Integrity:** An integral phrase must satisfy two conditions: (i) the multiple words in a phrase collocate together much more frequently than expected from random chance, and (ii) the phrase is a complete semantic unit, rather than a subsequence of another equally-frequent phrase.
- **Popularity:** A phrase is popular if it has a large number of occurrences. Representative phrases for a cell, in particular, should appear with some frequency within the documents of that cell. Very low frequency phrases within a cell do not contribute substantially to its semantics and so are not considered representative.
- **Distinctiveness:** High-popularity phrases that appear in many different cells constitute background noise, e.g., ‘earlier this month’ and ‘focus on’. Representative phrases should distinguish the target cell from its context, therefore provide more salient information to help users filter the noise. Distinctiveness is particularly critical in text cube scenarios, since analysts often navigate through the whole collection to find subsets of interest. Non-distinctive phrases will appear in many cells and offer redundant information.

However, none of the previous work has followed all three criteria. MCX [12, 1] follows *distinctiveness* (in a rough sense that only compare to the entire corpus) and ignores *popularity* and *integrity*. SegPhrase [5] addresses *integrity* in global quality phrase mining, but the notion of *popularity* and *distinctiveness* with respect to a target cell is not applicable to that problem setting. This paper proposes a new measure to evaluate all three criteria.

Within the whole ranked phrase list, top- k representative phrases normally have higher value for users in text analysis. As a further matter, the top- k query also enjoys computational superiority, so that users can conduct fast analysis. For these reasons, we define the problem as follows.

Definition 3 (Multi-Dimensional, Phrase-Based Summarization in Text Cube): Given a multi-dimensional text cube $\mathcal{TC} = (\mathcal{A}_1, \mathcal{A}_2, \dots, \mathcal{A}_n, \mathcal{DOC})$, it takes $c = (a_1, \dots, a_n, \mathcal{D}_c)$ as a query, and outputs top- k representative phrases based on the integrity, popularity and distinctiveness criteria.

Table 1: Top-10 representative phrases for NYT queries

⟨US, Gun Control⟩	⟨US, Immigration⟩	⟨US, Domestic Politics⟩	⟨US, Law and Crime⟩	⟨US, Military⟩
gun laws	immigration debate	gun laws	district attorney	sexual assault in the military
the national rifle association	border security	insurance plans	shot and killed	military prosecutors
gun rights	guest worker program	background check	federal court	armed services committee
background check	immigration legislation	health coverage	life in prison	armed forces
gun owners	undocumented immigrants	tax increases	death row	defense secretary
assault weapons ban	overhaul of the nation’s immigration laws	the national rifle association	grand jury	military personnel
mass shootings	legal status	assault weapons ban	department of justice	sexually assaulted
high capacity magazines	path to citizenship	immigration debate	child abuse	fort meade
gun legislation	immigration status	the federal exchange	plea deal	private manning
gun control advocates	immigration reform	medicaid program	second degree murder	pentagon officials

First, we acknowledge that these three criteria can all be subjective and relative, and it is difficult to find a clear binary judgment whether each phrase satisfies all the criteria. Therefore, we decide to use a score between 0 and 1 to characterize the degree of each phrase in satisfying these criteria. For phrase p in cell c , we use $int(p, c) \in [0, 1]$, $pop(p, c) \in [0, 1]$, and $disti(p, c) \in [0, 1]$ to denote the three criteria, and $r(p, c)$ to denote the overall ranking score that combines these criteria.

To combine the above criteria, we first notice that they reflect conjunctive conditions that should be satisfied, and one cannot replace the other. For example, popular word sequences may have quite low distinctiveness and sometimes ill-formed surface (*i.e.*, low integrity). Rare phrases that only occur once can be well distinctive. Since every criterion is indispensable, any low score (*i.e.*, near 0) in $int(p, c)$, $pop(p, c)$ or $disti(p, c)$ should result in a low rank for phrase p . Therefore, we design $r(p, c)$ as the geometric mean of those three scores.

The three criteria are equally positioned, though one can assign different weights according to user’s requirement in different applications. If one of the factors is close to 0, the geometric mean will be close to 0 as well. Alternatively, one can use harmonic mean to have the same property, but the score will then be strongly dominated by the weakest factor, which may be unfavorable because the role of the other two factors will be neglected.

When we design the concrete measures for each criterion, we are aware that the input documents can be any textual word sequences with arbitrary lengths, such as articles, titles, queries, tags, memos, messages and records. A good design of the measures should generalize well to a variety of text data. Therefore, we tend to use more statistical features and fewer linguistic features.

Now we discuss design principles that are more specific to the three criteria.

- Popularity and distinctiveness of a phrase are dependent of the target cell, while integrity is not. Hence, $int(p, c)$ can be simplified as $int(p)$.
- Popularity and distinctiveness can be measured from frequency statistics of a phrase in each cell, while integrity cannot. To measure integrity, one needs to investigate each occurrence of the phrase and other phrases to determine whether that phrase is indeed an integral semantic unit. We leverage *SegPhrase* [9] to compute integrity.
- Popularity relies on statistics from documents only within the cell \mathcal{D}_c , while distinctiveness relies on documents both in and out of the cell. We define the documents involved for distinctiveness measure calculation as *contrastive document set*. More precise distinctiveness measure requires appropriate choice of contrastive document set. In our particular algorithm design, sibling set $\mathbb{S}(c)$ is used as contrastive document set.

With the phrase ranking algorithm designed based on the aforementioned principles, it is applied on NYT 2013-2016 dataset and PubMed Cardiac data for quality evaluation. Our algorithm is referred as **RepPhrase** and multiple baselines are referred as **MCX** [12], **SegPhrase** [9] and their combinations.

Case study on NYT. We show 5 real queries in NYT dataset and their representative phrase list in Table 1. Query ⟨US, Gun Control⟩ and ⟨US, Immigration⟩ are siblings, ⟨US, Domestic Politics⟩ is their parent cell. ⟨US,

Table 2: Top representative phrases for 5 cardiac diseases

⟨Cerebrovascular Accident⟩	⟨Ischemic Heart Disease⟩	⟨Cardiomyopathy⟩	⟨Arrhythmia⟩	⟨Valve Dysfunction⟩
alpha-galactosidase a	Cholesteryl ester transfer protein	Interferon gamma	Methionine synthase	Mineralocorticoid receptor
brain neurotrophic factor	apolipoprotein a-I	interleukin-4	ryanodine receptor 2	tropomyosin alpha-1 chain
tissue-type activator	integrin alpha-iib	interleukin-17a	potassium v.g. h member 2	elastin
apolipoprotein e	adiponectin	titin	inward rectifier channel 2	beta-2-glycoprotein 1
neurogenic l.n.h.p. 3	p2y purinoceptor 12	tumor necrosis factor	beta-2-glycoprotein 1	myosin-binding protein c

Domestic Issues), ⟨US, Law and Crime⟩ and ⟨US, Military⟩ are also siblings. For the first two queries, the discovered phrases are specific to gun control and immigration. There are both entity names like *the national rifle association* and *guest worker program* and event-like phrases like *assault weapons ban* and *overhaul of the nation's immigration laws*. In their parent cell ⟨US, Domestic Politics⟩, the top phrases cover various children cell topics, including gun control, immigration, insurance act and federal budget. This list provides very informative phrases that describe the major content. For the two siblings of ⟨US, Domestic Politics⟩ (last two columns), the lists also cover the main entities involved and the major topics, e.g., *second order murder*, *sexual assault in the military*, etc.. Also notice that, these top lists of representative phrases keep good balance between short phrases and long phrases. That is mainly credited the consideration of both popularity and distinctiveness without introducing bias to phrase length.

Case study on PubMed Cardiac data. In collaboration with UCLA BD2K team, we apply the phrase-based summarization on PubMed cardiac publications. They provide 5 categories of cardiac diseases and a set of 300+ protein candidates. The goal of our summarization is to discover top contributing proteins for these disease categories. The results are shown in Table 2. These top proteins help medical scientists find more concrete direction to look into and largely reduce the time spent on reading irrelevant publications. Since the *distinctiveness* is a major criterion in our ranking, we note that non-informative proteins that are related to all diseases, like *amyloid beta a4 protein*, are not included in the top list. Another exciting discovery is that protein *titin*, a newly discovered protein, is also listed as top protein for *Cardiomyopathy*.

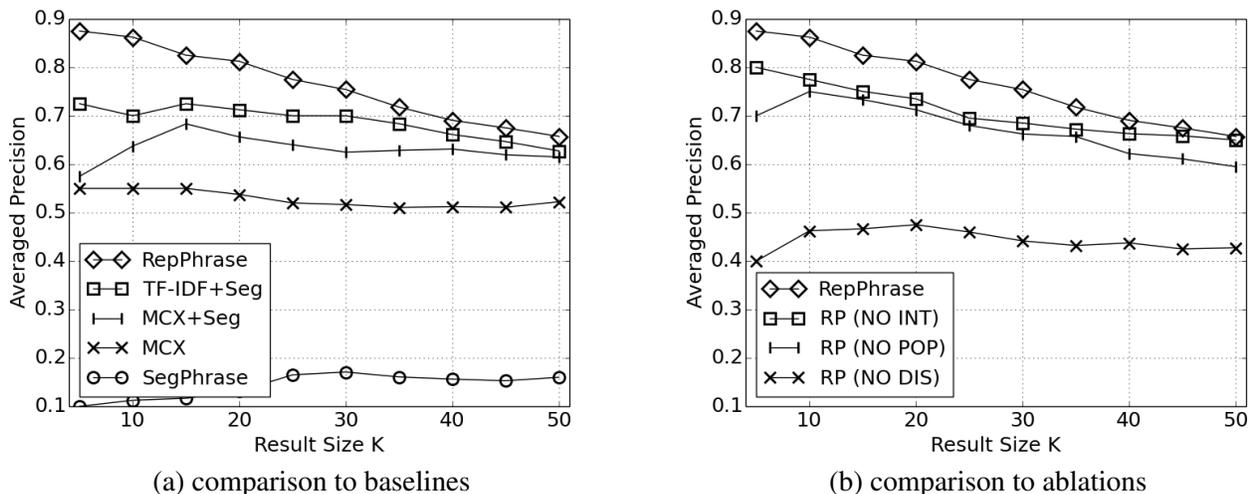


Figure 5: Phrase assignment accuracy

Phrase-to-cell assignment accuracy. The idea of this experiment is to quantify how many phrases among top- k results of a cell indeed represent the semantics of that cell. We test eight queries. Four of them are *1-Dim Queries*, and the other four are *2-Dim Queries*. To generate non-trivial test queries, we first randomly pick two *1-Dim Queries* and two *2-Dim Queries*; then for each picked query, we add the most similar sibling in terms

of both size and content as a paired query. To ease the labeling, for each pair of test queries, we first collect all top-50 phrases generated by all the measures for both queries. For each phrase in the pool, we label it with either one of the two cells which it best represents, or a ‘None’ label in three circumstances: 1) it is not a valid phrase, 2) it is not relevant to either cell and 3) it is a background phrase that are shared by both cells. We then measure the accuracy of phrase assignment by the average precision from top-5 to top-50 phrases. We show the result in Figure 5(a) for baselines and Figure 5(b) for ablations.

In general, as k grows, the precisions of those measures go down. In Figure 5(a), **RepPhrase** has the best precision and **SegPhrase** has the worst. Also, the difference of precision between **RepPhrase** and others decreases as k grows. That is attributed to the limited number of true representative phrases. **RepPhrase** successfully ranks these good phrases high, others gradually include them as k grows. Amongst all the baselines, **TF-IDF+Seg** outperforms others since it is the only baseline that captures all three criteria. However, it still loses to **RepPhrase**. Both use sibling cells as contrastive group, using classification probability (**RepPhrase**) as *distinctiveness* performs better than using IDF (**TF-IDF+Seg**).

In Figure 5(b), we show the performance drop by removing one of the three criteria respectively. We notice that **RP (NO INT)** has the best precision amongst all ablations and **RP (NO DIS)** has the worst, which indicates the relative importance of the criteria: *distinctiveness* > *popularity* > *integrity*. One interesting comparison is between **MCX+Seg** and **RP (NO POP)**. These two can be viewed as two versions of standalone *distinctiveness* measure with different contrastive document groups. Using dynamic sibling cells as contrastive group (**RP (NO POP)**) performs better than using the static entire collection (**MCX+Seg**), especially on the top phrases. It further justifies the choice of using dynamic background over static background.

3 Platform for multi-dimensional text analysis

As discussed above, converting text corpora into multi-dimensional text cubes provides various benefits, including i) flexibility of user queries that captures insights with different granularities and ii) contextualized analysis that is able to discover comparative insights. To support such general cube-based analytical tasks, we proposed and implemented the generalized infrastructure. Like the phrase-based summarization task, other multi-dimensional analytical tasks share similar computational and operational characteristics.

Computational Characteristic: given a pre-defined dimension structure, similar to traditional *OLAP*-operations, proper pre-computation (called *materialization*) helps to speedup online user queries and therefore supports real-time query responses. Number of possible multi-dimensional queries are often exponential, thus it is necessary to intelligently select partial cells to materialize. Since these text analytical measures are more complicated than traditional *distributive* and *algebraic* measures, we normally need to materialize intermediate result and require reasonable amount of online computation after query is issued. Such hybrid computational scheme is normally shared by various analytical tasks.

Operational Characteristic: different analytical tasks share the same input/output format. Normally, the system takes a multi-dimensional user query as input, i.e., $\langle \text{China, Economy} \rangle$, and returns structured textual result, i.e., top- k phrases in phrase-based summarization and k -topics in cube-based topic modeling. Therefore, many operations including *indexing*, *retrieval* and etc. can also be shared by different tasks.

Therefore, we created a platform for general multi-dimensional text analysis. It provides a generalized platform that can easily import any collection of free text and structured data, such as news data, aviation reports or academic papers, extract entities, construct the text-rich data cube and support powerful search and mining functions. For structured data, multidimensional data cube can be constructed easily. For text-intensive data with minimally predefined structured information (e.g., news data), natural language and information extraction tools can be used to extract entities of multiple types such as person, location, organization, time, and event. This platform provides a tremendous opportunity to conduct multi-dimensional analysis on text and structured data in powerful and flexible ways.

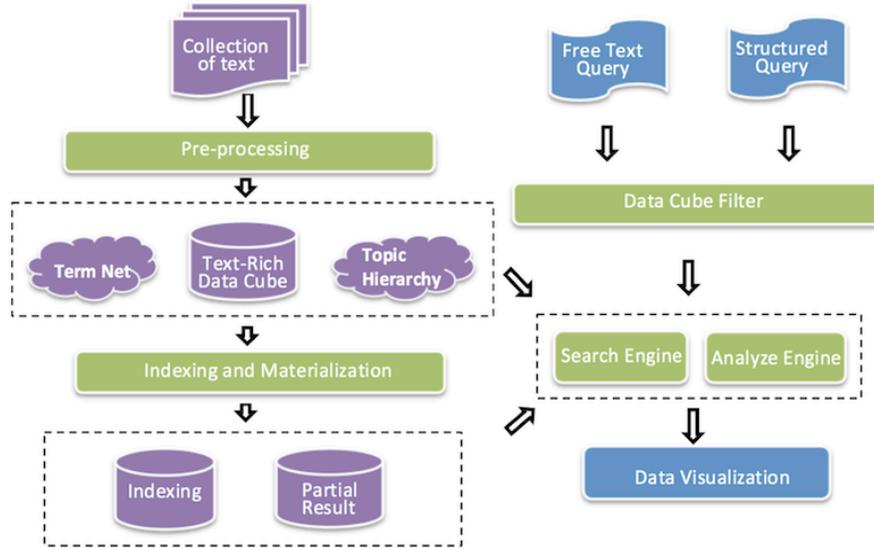


Figure 6: System Architecture of the Platform

System Architecture. The platform is designed as shown in Figure 6. It consists of the following modules: (1) *Data Uploading and Preparation*, which pre-processes the free text corpus from user’s uploading and converts it into a text-rich data cube with term network and topic hierarchy extracted; (2) *Indexing and Materialization*, which builds indexing and partial materialization results for keyword search, top cell finding, single dimension distribution and hierarchical topic modeling; (3) *Query-Based Search and Mining Module*, which processes user-queries (both search and analysis queries) by parsing the query, selecting and executing appropriate search or mining module (which searches or mines on the constructed text-rich data cube to derive results); and (4) result presentation by *Visualization and Interpretation* of the search/mining processes and results.

The platform reveals another advantage of converting text corpora into multi-dimensional text cubes, that is the power of real-time text analysis. The rich structure embedded in text cubes empowers smart indexing and materialization that enables real-time processing of any multi-dimensional query. Without such multi-dimensional structure, it is challenging to support real-time text analysis on arbitrary portion of large text corpora.

3.1 Real-time Phrase-based Summary Generation

In this section, we use phrase-based summarization as example task to introduce how the offline/online computation scheme is implemented.

Utility-Guided Materialization. In offline computation, we extend the *GreedySelect* algorithm [8] to our task and develop the utility-guided partial materialization. The algorithm first conducts a *topological sorting* by the *parent-descendant* relationship in the cube space. Then it traverses the cells in the bottom-up order. This order ensures that all cells used for aggregating the current cell must have been examined, so the dynamic programming of cost estimation can proceed. For each cell, we do not simply materializing all the required cells (all siblings). Instead, it repeatedly attempts materialization of one sibling, and reevaluates the cost of querying the target cell, until it falls below threshold. The order of choosing siblings affects how many siblings will be materialized and how much storage cost is needed to meet the constraint. We use a *utility* function for each sibling cell c' to guide this process.

Optimized Online Processing. The vanilla online processing needs to compute the ranking measure for all phrase candidates in a cell in order to sort them. The computation of the distinctiveness score can be expensive,

if the cell is not materialized. We propose an early termination and skipping technique to prune phrase candidates that are impossible to be among top- k .

We evaluate the computational performance using the full NYT dataset **4-Dim Cube** and **6-Dim Cube** (both have 4.7 million articles, 17.04 GB raw size, but different dimension numbers). For the offline computation, we compare the following algorithms for materializing phrase-level statistics: 1) **FULL** (full materialization), 2) **LEAF** (leaf materialization), 3) **GREEDY** (in [8]) and **UTILITY 1-5** (with 5 different utility functions).

Table 3: Space-time trade-off of **LEAF** and **FULL**

	4-Dim Cube		6-Dim Cube	
	Space (GB)	Time (s)	Space (GB)	Time (s)
LEAF	0.68	73.2	26.76	3407.5
FULL	20.17	0.86	706.0	0.89

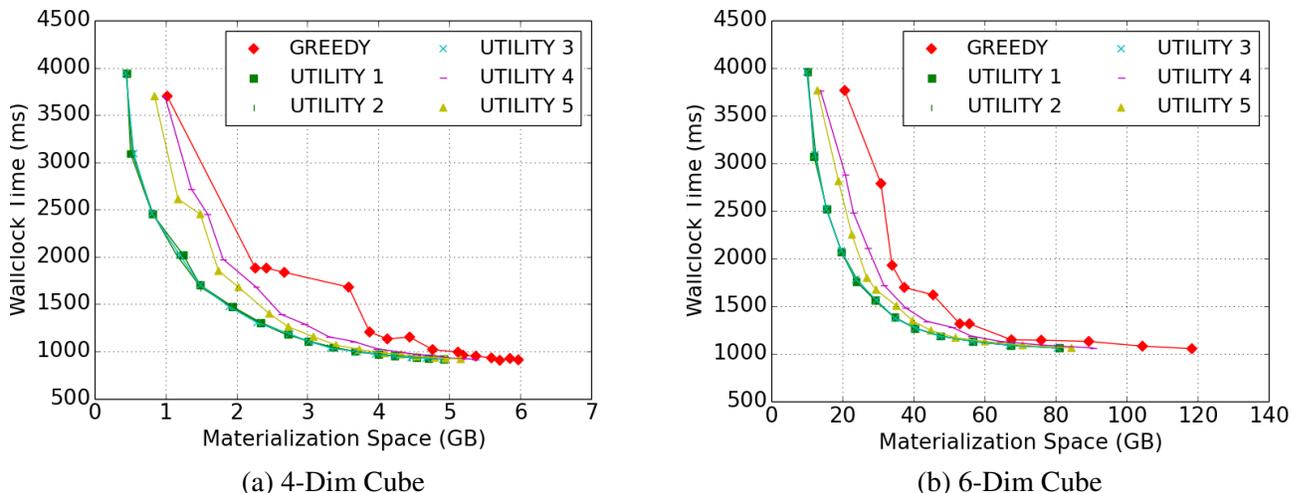


Figure 7: Time-space balance

Figure 7(a) and 7(b) shows the space-time trade-off on **4-Dim Cube** and **6-Dim Cube**. Since **LEAF** and **FULL** strategies have quite exceptional worst query time or materialization space, the result is separately shown in Table 3. We first notice that the space cost of **LEAF** is as low as 26.76 GB in **6-Dim Cube**, but the worst query time is more than 3,400 seconds. If we materialize every cell as in **FULL**, it has the minimized worst query time but consumes about 706 GB to materialize. The other 6 strategies make trade-offs between time and space by setting different latency constraint. We notice that all five utility-guided strategies outperform **GREEDY**, *i.e.*, their curves are closer to the origin point. In particular, picking any of **UTILITY 1-3** yields the best trade-off that can take less than 10% of the storage compared to **FULL** and less than 50% of the **GREEDY** strategy with same worst query time.

4 Conclusion

This paper proposes multi-dimensional text analysis in text cubes and an interesting application: multi-dimensional phrase-based summarization. It mines top- k representative phrases based on three criteria: integrity, popularity and distinctiveness. We propose a fine-grained distinctiveness assessment that considers phrase distributions across sibling cells. This is shown to be more effective than previous measures. Given computational

challenges imposed by these textual measures, we develop a generalized platform to support efficient online and offline computational optimization. These can be generally applied to any measure in text cubes.

There are several possible extensions of the current problem to explore in future work. (1) Instead of outputting top- k phrases, one can design measures for generating top- k semantic clusters, which improve coverage of the content and reduce semantic redundancy. (2) Users may make a sequence of OLAP queries before navigating to the target cell. One can study the patterns of such query sequence and develop semantic representations accordingly. (3) One can further investigate the context-aware materialization problem. It will be useful to develop algorithms with stronger theoretical guarantee in optimizing the time-space trade-off.

Beside phrase-based summarization, other useful text analytical problems can be studied within data cube scenarios, including outlier detection, sentence-based summarization, and sentiment analysis. We believe the multi-dimensional framework can help us achieve real-time, flexible and contextualized analysis for such tasks.

References

- [1] S. Bedathur, K. Berberich, J. Dittrich, N. Mamoulis, and G. Weikum. Interesting-phrase mining for ad-hoc text analytics. *PVLDB*, 2010.
- [2] O. Ben-Yitzhak, N. Golbandi, N. Har'El, R. Lempel, A. Neumann, S. Ofek-Koifman, D. Sheinwald, E. Shekita, B. Sznajder, and S. Yogev. Beyond basic faceted search. In *WSDM*, 2008.
- [3] D. Dash, J. Rao, N. Megiddo, A. Ailamaki, and G. Lohman. Dynamic faceted search for discovery-driven analysis. In *CIKM*, 2008.
- [4] B. Ding, B. Zhao, C. X. Lin, J. Han, and C. X. Zhai. Topcells: Keyword-based search of top-k aggregated documents in text cube. In *ICDE*, 2010.
- [5] A. El-Kishky, Y. Song, C. Wang, C. R. Voss, and J. Han. Scalable topical phrase mining from text corpora. *PVLDB*, (3), 2014.
- [6] M. A. Hearst. Clustering versus faceted categories for information exploration. *CACM*, (4), 2006.
- [7] A. Inokuchi and K. Takeda. A method for online analytical processing of text data. In *CIKM*, 2007.
- [8] C. X. Lin and e. a. Ding, Bolin. Text cube: Computing ir measures for multidimensional text database analysis. In *ICDM*, 2008.
- [9] J. Liu, J. Shang, C. Wang, X. Ren, and J. Han. Mining quality phrases from massive text corpora. In *SIGMOD*, 2015.
- [10] J. M. Pérez-Martínez, R. Berlanga-Llavori, M. J. Aramburu-Cabo, and T. B. Pedersen. Contextualizing data warehouses with documents. *Decision Support Systems*, 45(1):77–94, 2008.
- [11] F. Ravat, O. Teste, R. Tournier, and G. Zurfluh. Top_keyword: an aggregation function for textual document olap. In *Data Warehousing and Knowledge Discovery*. Springer, 2008.
- [12] A. Simitsis, A. Baid, Y. Sismanis, and B. Reinwald. Multidimensional content exploration. *PVLDB*, (1), 2008.
- [13] F. Tao, G. Brova, J. Han, H. Ji, C. Wang, B. Norick, A. El-Kishky, J. Liu, X. Ren, and Y. Sun. Newsnetexplorer: automatic construction and exploration of news information networks. In *SIGMOD*, 2014.
- [14] F. Tao, J. Han, et al. Eventcube: multi-dimensional search and mining of structured and text data. In *KDD*, 2013.
- [15] D. Tunkelang. Faceted search. *Synthesis lectures on information concepts, retrieval, and services*, (1), 2009.
- [16] B. Zhao, X. Lin, et al. Texplorer: keyword-based object search and exploration in multidimensional text databases. In *CIKM*, 2011.