OLAP Textual com Múltiplas Hierarquias de Tópicos e Rankings Segmentados

Adriano Neves P. Souza
Dep. de Ciência da Computação
Universidade Federal de Ouro Preto
Ouro Preto - MG - Brasil
adrianonevesps@gmail.com

Reinaldo Silva Fortes
Dep. de Ciência da Computação
Universidade Federal de Ouro Preto
Ouro Preto - MG - Brasil
reifortes@iceb.ufop.br

Joubert de Castro Lima
Dep. de Ciência da Computação
Universidade Federal de Ouro Preto
Ouro Preto - MG - Brasil
joubert@iceb.ufop.br

RESUMO

Na última década a tecnologia OLAP tem sido redesenhada para melhor atender à demanda de dados textuais, tendo que remodelar suas medidas e dimensões. A hierarquia de tópicos surgiu como uma alternativa para organizar dimensões textuais em diferentes níveis semânticos. Contudo, tal hierarquia é criada uma única vez e utilizada para todos os cuboides do cubo. A hierarquia de tópicos é sensível ao conteúdo dos documentos, portanto diferentes células de um cubo agregam diferentes conjuntos de documentos, produzindo hierarquias de tópicos distintas. Este artigo apresenta uma abordagem para OLAP textual que constrói múltiplas hierarquias de tópicos para cada célula do cubo, denominada DTCubing. Múltiplas hierarquias são viáveis porque cada documento pode ser particionado em diversos segmentos de texto, tais como título, resumo, parágrafo, dentre outros. Este artigo também pretende contribuir com a apresentação dos resultados das consultas multidimensionais. O estado da arte em OLAP textual normalmente retorna os top-k documentos mais relevantes como resultado de suas consultas. A abordagem DTCubing vai além, retornando também os top-k segmentos de texto mais relevantes, portanto os parágrafos e resumos mais relevantes podem ser retornados. Os experimentos realizados utilizando artigos indexados pela DBLP confirmam as hipóteses do trabalho.

Palavras-chave

Cubo de Dados, OLAP, OLAP textual, Base de dados textual, Ranking, Hierarquia de Tópicos.

ABSTRACT

In the last decade, the OLAP technology has been redesigned for the textual data, therefore dimensions, hierarchies and measures are being remodeled. Topic hierarchy is a useful alternative to organize document collections. Currently, the topic hierarchy is defined only once in the data cube, i.e., for the entire lattice of cuboids. However, textual hierarchy is sensitive to the content of the documents. Thus, a data cube cell can contain a collection of documents distinct from others in the same cube, since they have complementary aggregation levels that potentially introduce changes in the topic hierarchy. In this paper, we present a textual OLAP approach, named DTCubing, which handles multiple topic hierarchies for each cube cell. Multiple hierarchies are feasible because a document can be partitioned into several text segments (e.g., title, abstract, keywords and many more). A second contribution of this paper refers to query response. The state of art in textual OLAP normally returns the top-k documents as a query result. We go beyond by returning other text segments, such as the most significant titles, abstracts and paragraphs. Experiments, using part of the DBLP papers, reinforce our assumptions.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
SBSI 2017, June 5th–8th, 2017, Lavras, Minas Gerais, Brazil.
Copyright SBC 2017.

CCS Concepts

• Information systems → Data management systems → Database management system engines → Online analytical processing engine.

Keywords

Textual OLAP, data cube, textual data, topic hierarchy, document ranking, text segment.

1. INTRODUÇÃO

A popularização de redes sociais (*Facebook*, *Twitter* e *Instagram*) e agregadores multimídia (*Youtube* e *Wikipedia*) têm provocado um enorme aumento na quantidade de dados não estruturados [4]. Este cenário dificulta muito o trabalho de gestores, executivos e analistas, uma vez que, segundo [20], a quantidade excessiva de documentos excedeu em muito a capacidade humana de compreensão dos dados. Estes obstáculos de análise iniciam a nossa demanda para a remodelagem das ferramentas de *Online Analytical Processing* (OLAP) tradicionais, uma vez que as mesmas se mostram inadequados para o tipo de dado textual [2].

O operador cubo de dados [6], núcleo de qualquer ferramenta OLAP, enfrenta novos desafios ou impossibilidades quando aplicado à bases de dados multidimensionais textuais. Suas hierarquias normalmente são modeladas por algum especialista e apenas uma vez durante todo o ciclo de vida do cubo. Além disso, não são úteis neste novo contexto, pois não consideram nenhuma relação semântica e nem a dinâmica dos níveis hierárquicos da coleção de documentos.

Hierarquias são fundamentais no processo decisório, pois permitem o mapeamento de conceitos de baixo nível em conceitos de alto nível [6]. Desta forma reduzem o espaço de busca, agilizando e simplificando a análise do usuário. É intuitivo perceber que o atributo *ano* é hierarquicamente superior ao atributo *dia*, mas quando o atributo é uma coleção de documentos, como é possível detectar se um termo, um segmento de texto ou mesmo todo o documento é hierarquicamente superior ou inferior a outro? Diferentes abordagens de cubos de dados textuais foram propostas para resolver o desafio de se navegar através de dados estruturados e dados textuais de forma integrada [8–11, 13–15, 19, 21–23].

A literatura em OLAP textual que suporta o conceito de hierarquia de tópicos ainda sofre sérias limitações, uma vez que assumem a

criação de tal hierarquia uma única vez e para toda a coleção de documentos. Desta forma, um tópico t_1 é sempre hierarquicamente superior a um tópico t_2, independente da célula do cubo selecionada. A primeira hipótese deste trabalho assume que não basta construir uma única hierarquia de tópicos a partir de toda a coleção de documentos, mas sim criar hierarquias de tópicos dinâmicas, a partir de subconjuntos de documentos, obtidos à medida em que filtros são aplicados a uma consulta. Tal estratégia permite que o usuário desfrute de uma maior variedade de tópicos, direcionando sua pesquisa e obtendo conjuntos de documentos cada vez mais relevantes ao contexto de sua consulta.

Os resultados das consultas na OLAP textual são os documentos mais relevantes, ranqueados com diferentes métodos [13–15, 22, 23]. A segunda hipótese deste trabalho assume que muitas consultas objetivam retornar segmentos de texto mais específicos, como *parágrafos*, *resumos*, *capítulos*, *palavras-chave*, *sentenças* e outros. Suponha uma consulta retornando segmentos de texto relevantes ao tópico "*Big Data*". O usuário deveria poder definir se almeja visualizar a lista dos *parágrafos*, *sentenças, capítulos* ou mesmo a lista dos *documentos* mais relevantes a tal tópico. Podemos notar que existe uma grande diferença nos rankings retornados, uma vez que as métricas propostas na literatura são sensíveis ao número de termos, portanto aplicá-las a um texto com poucos ou muitos termos, que possuem uma alta correlação com um determinado tópico, altera o resultado substancialmente, conforme indicam os experimentos realizados.

Neste artigo é apresentada uma nova abordagem para OLAP textual, denominada *Dynamic Topic Cubing* ou DTCubing abreviadamente, que propõe a construção de hierarquias de tópicos dinâmicas e múltiplas. Dinâmicas por que são criadas a cada nova célula obtida do cubo textual e múltiplas por serem construídas a partir de diferentes segmentos de texto (*título*, *resumo*, *parágrafo*, etc.). Como consequência, DTCubing oferece múltiplos rankings como medidas de suas consultas, de acordo com o segmento de texto escolhido. Avaliações experimentais confirmam tais hipóteses.

Algumas conferências que compõem a DBLP[1] foram utilizadas para a condução dos experimentos. Um novo algoritmo baseado na abordagem qCube [16] foi implementado para computar tanto as múltiplas hierarquias de tópicos quanto os múltiplos rankings de segmentos.

O restante deste trabalho está organizado da seguinte forma: a Seção 2 apresenta o conjunto de trabalhos relacionados utilizados como base para o desenvolvimento da abordagem DTCubing. Na Seção 3, é apresentada a definição formal da abordagem DTCubing. Na Seção 4, são evidenciadas as quatro etapas que constituem a abordagem DTCubing. A Seção 5 apresenta a análise experimental que sustenta a utilidade da abordagem. Na seção 6, são apresentadas as conclusões, finalizando as discussões do trabalho.

2. TRABALHOS RELACIONADOS

Todas as abordagens da literatura foram projetadas para a construção de uma única hierarquia para a dimensão textual, sendo ela utilizada por todo o *lattice* de cuboides. Tais abordagens podem ser classificadas de acordo com a construção de tal hierarquia. Diversos trabalhos apresentam uma construção manual para a hierarquia da dimensão textual, utilizando especialistas ou fontes externas de conhecimento, como *WordNet*[2] ou qualquer outro repositório de informação. Outras adotam estratégias para construção automática de tal hierarquia.

Hierarquias Textuais Manuais: Text Cube [13] foi o primeiro trabalho a propor uma hierarquia para a dimensão textual, nomeada hierarquia de termos. A hierarquia de termos é representada na forma de uma árvore, onde cada nó folha representa um único termo e o nó raiz corresponde à junção de todos os termos da coleção de documentos. Os nós intermediários são organizados de acordo com a semântica dos níveis inferiores. Ferramentas como *WordNet* podem ser utilizadas no processo de agregação dos termos semanticamente semelhantes, contudo a topologia da árvore é determinada manualmente.

Topic Cube [23] implementa um aperfeiçoamento da abordagem Text Cube, utilizando o conceito de tópicos para a construção da hierarquia textual. Agora a hierarquia não agrega mais todos os termos da coleção de documentos. Ao invés disto, são usados apenas os tópicos em que os usuários podem estar mais interessados, oferecendo uma representação mais simples e precisa, nomeada hierarquia de tópicos. Topic Cube incorpora uma análise probabilística com o modelo PLSA (*Probabilistic Latent Semantic Analysis*) para calcular a probabilidade de cada documento pertencer a um nó da hierarquia de tópicos previamente definida.

Cube Index [8] e [9], adota os conceitos da abordagem Text Cube para a implementação de três índices denominados: *direct index*, *next word index* e *inverted index*. A hierarquia de documentos apresentada em [9] não especifica nenhuma relação semântica entre um termo e um documento, sendo apresentada na forma de uma árvore. Sua topologia possui exatamente cinco níveis, classificados como: (i) palavra; (ii) par de palavras; (iii) sentença; (iv) parágrafo; (v) documento.

Em [15], é apresentada uma abordagem de cubo de dados textual baseada em contextos e denominada CXT-Cube. Nela, cada dimensão está relacionada a um fator contextual, apresentando um método de propagação de relevância que permite alterar valores dos nós folhas de uma hierarquia de tópicos, construída de forma manual e extraída de fontes externas de conhecimento.

Diversas abordagens foram propostas na literatura com o intuito de aperfeiçoar as medidas da OLAP textual. Em [5] é apresentada uma solução para o problema de encontrar as *top-k* células mais relevantes em um Text Cube, reduzindo o espaço de busca com a estimativa de limites superiores de relevância. Assim, há como explorar o menor número de células para responder a uma consulta.

Em [22] há uma abordagem, denominada MicroTextCluster Cube (ou MiTexCube), que inclui o conceito de "*micro clusters*" de documentos como uma representação compacta de toda a coleção, permitindo agrupar os documentos de cada célula utilizando a relevância entre eles.

Os trabalhos [14] e [11] estendem a abordagem Text Cube, construindo medidas mais especializadas às suas áreas de interesse, como medidas HSCB (*Human Social Cultural Behavior*), tf-idf (*term frequency- inverted document frequency*) e LM (*Language Model*).

Hierarquias Textuais Automáticas: As abordagens iNextCube [21] e EventCube [19] implementam tanto as medidas textuais propostas em Text Cube quanto as de Topic Cube, reunindo

[1] http://dblp.org

[2] http://wordnet.cs.princeton.edu/

portanto, medidas de Recuperação de Informação (RI) e medidas probabilísticas (PLSA). iNextCube adota o método *NetClus* [17] na construção automática de suas hierarquias.. A hierarquia apresentada para a dimensão textual utilizou artigos da DBLP, tendo os níveis hierárquicos conhecidos previamente e extraídos de uma fonte externa de conhecimento. A associação entre os níveis hierárquicos é construída automaticamente, utilizando análise de redes de informação. EventCube se diferencia da abordagem iNextCube por utilizar o método CATHY [3] para a construção de suas hierarquias, não exigindo o auxílio de especialista ou fonte externa de conhecimento.

A abordagem DTCubing se diferencia de todos os trabalhos apresentados anteriormente, implementando a construção dinâmica de múltiplas hierarquias de tópicos para cada célula do cubo de dados. Soma-se a isso o fato de também apresentar os *top-k* segmentos de texto mais relevantes, uma vez que a literatura apresenta apenas os documentos mais relevantes como resultado de suas consultas multidimensionais.

3. DTCUBING

Nesta seção a abordagem DTCubing é detalhada. A seção 3.1 apresenta uma formalização detalhada de um segmento de texto. A seção 3.2 apresenta a definição formal de um cubo DTCubing, assim como suas células e medidas. A seção 3.3 detalha as hierarquias construídas em DTCubing e a definição formal de suas estruturas.

3.1 Segmentos de Texto

Um segmento de texto pode ser formalizado como qualquer conjunto de termos subsequentes pertencentes a um documento, seguindo um padrão para a sua estruturação. Todo segmento de texto é formado por um conjunto de termos, sendo assim, é possível observar uma relação de equivalência entre dois conjuntos de segmentos, onde a união dos termos de cada conjunto separadamente, corresponde ao mesmo conteúdo final. A união de todos os parágrafos de um documento, por exemplo, é equivalente à união de todas as sentenças do mesmo documento, uma vez que ambas as segmentações levam ao mesmo conjunto final de termos, já que toda sentença de um documento está contida em algum parágrafo do mesmo documento.

3.2 Conceitos

Um cubo de dados é composto de células base e células agregadas. Uma célula multidimensional agregada contém pelo menos um valor de atributo de dimensão igual a ALL(*). Uma célula base contém apenas valores de atributos diferentes de ALL. O *wildcard* ALL representa todos os valores de um atributo de dimensão específico.

DTCubing é composto de dimensões estruturadas, hierarquias de tópicos e medidas, apresentadas como rankings de segmentos. Células base e células agregadas também são encontradas em DTCubing, porém o valor ALL é adotado apenas em dimensões estruturadas, produzindo um *lattice* de cuboides da mesma forma que um cubo de dados estruturado tradicional. Computações *bottom-up*, *top-down* ou híbridas podem ser utilizadas da mesma forma que ocorrem em OLAP não textual.

Um cubo DTCubing pode ser formalizado como $DTC = \{D_1, D_2, \dots, D_n, T, R\}$ onde D_i representa a i-*ésima* dimensão no cubo de dados. Uma dimensão D_i é composta de múltiplas hierarquias, então $D_i = \{\{A_1, A_2, A_4, \dots\}, \{A_3, A_5, \dots\}, \dots, \{A_1, A_{34}, \dots\}\}$ ou $D_i = \{A_1, A_2, A_3, A_4, A_5 \dots, A_{34} \dots\}$, onde cada $\{A_i, A_j\}$ corresponde a uma hierarquia de uma dimensão estruturada de DTC (ex. a dimensão tempo pode ser organizada segundo as hierarquias $\{\{ano, mês\}, \{ano, mês, hora\}, \dots\}$. A dimensão $T = \{DOC, M\mathcal{H}_{\mathcal{S}_1}, M\mathcal{H}_{\mathcal{S}_2}, \dots, M\mathcal{H}_{\mathcal{S}_m}\}$ representa a dimensão textual, composta pela coleção de documentos DOC e todas as possíveis hierarquias de tópicos $\{M\mathcal{H}_{\mathcal{S}_1}, M\mathcal{H}_{\mathcal{S}_2}, \dots, M\mathcal{H}_{\mathcal{S}_m}\}$. $R = \{M\mathcal{R}_{\mathcal{S}_1}, M\mathcal{R}_{\mathcal{S}_2}, \dots, M\mathcal{R}_{\mathcal{S}_m}\}$ corresponde ao conjunto de rankings retornados por DTCubing, onde cada $M\mathcal{R}_{\mathcal{S}_i}$ corresponde a um ranking construído a partir do conjunto de segmentos $[\mathcal{S}_i]$. $[\mathcal{S}_i]$ representa o conjunto de segmentos do tipo *i*, que pode ser igual a *título*, *resumo*, *palavras-chave*, *parágrafo*, *sentença* entre outros.

Uma célula DTCubing pode ser representada como $C = (a_1, a_2, a_3, \dots, a_n, \mathcal{H}_{\mathcal{S}_1}, \mathcal{H}_{\mathcal{S}_2}, \dots, \mathcal{H}_{\mathcal{S}_m}, R)$, onde cada a_i corresponde ao valor de um atributo estruturado de uma dimensão. Cada a_i está associado a uma única dimensão. $\mathcal{H}_{\mathcal{S}_i}$ corresponde à hierarquia de tópicos construída para a célula C, a partir do conjunto $[\mathcal{S}_i]$ de segmentos de texto. $\mathcal{R}$ é a medida agregada na célula C, composta do conjunto $\{\mathcal{R}_{\mathcal{S}_1}, \mathcal{R}_{\mathcal{S}_2}, \dots, \mathcal{R}_{\mathcal{S}_m}\}$, onde $\mathcal{R}_{\mathcal{S}_i}$ é o i-*ésimo* ranking de segmentos construído a partir de $[\mathcal{S}_i]$. Neste exemplo, existem m tipos de segmentos por documento em DOC. Cada hierarquia de tópicos em DTCubing é estruturada na forma de uma árvore, onde cada nó corresponde a uma distribuição θ_j de tópicos. Sendo assim, $\mathcal{H}_{\mathcal{S}_i} = \{\theta_1, \theta_2, \dots, \theta_k\}$, onde θ_j representa a j-*ésima* distribuição de tópicos de $\mathcal{H}_{\mathcal{S}_i}$. Cada distribuição pode ser representada como $\theta_j = \{t_1, t_2, \dots, t_z\}$, onde cada t_i corresponde a um tópico elencado pela hierarquia.

3.3 Hierarquias de Tópicos

As hierarquias de tópicos em DTCubing são construídas de forma automática, sem a necessidade de especialistas ou fontes externas de conhecimento. Tais hierarquias auxiliam o usuário na tarefa de análise dos dados, uma vez que reduzem o espaço de busca por apresentarem os atributos da dimensão textual, ou seja, os documentos de texto organizados em diferentes níveis semânticos.

As hierarquias de tópicos apresentadas em DTCubing são construídas a cada nova célula retornada pelo cubo de dados. A partir do momento em que o usuário modifica o contexto de sua consulta, obtendo uma nova célula, apenas os documentos ali agregados são utilizados para a construção das hierarquias de tópicos, elencando tópicos muito mais específicos do que uma hierarquia global, construída para toda a coleção de documentos.

As abordagens OLAP textuais que apresentam hierarquias de tópicos automáticas utilizam apenas um segmento de texto no processo de construção. DTCubing apresenta uma nova estratégia, construindo múltiplas hierarquias de tópicos, a partir de diferentes segmentos de texto, a fim de apresentar uma maior variedade de tópicos ao usuário. O número de hierarquias por célula depende do tipo de consulta executada pelo usuário. Seja c_i uma célula específica do cubo de dados e $\mathcal{S}_D = \{[\mathcal{S}_1], [\mathcal{S}_2], \dots, [\mathcal{S}_m]\}$ o conjunto de segmentos associados a tal célula. Se c_i foi obtida a partir de uma consulta $q_1 = (Conferência = \text{IJCAI}, Ano = 2010)$, contendo filtros apenas nas dimensões estruturadas (*Conferência* e *Ano*, respectivamente), DTCubing constrói uma hierarquia de tópicos para cada $[\mathcal{S}_i] \in \mathcal{S}_D$ que não possua relação de equivalência com nenhum outro conjunto de segmentos. Para aqueles conjuntos que possuem relação de equivalência, apenas um conjunto é escolhido, originando mais uma hierarquia de tópicos.

Como exemplo, não se computaria a hierarquia de tópicos a partir do segmento *sentença*, pois já se computou a hierarquia a partir dos *parágrafos*. As segmentações por *parágrafo* e por *sentença* resultam no mesmo conjunto final de termos, assim como explicado

na seção 3.1, resultando em hierarquias muito similares. Formalmente, o conjunto $\mathcal{H}_{c_i} = \{\mathcal{H}_{\mathcal{S}_1}, \ldots, \mathcal{H}_{\mathcal{S}_k}\}$ corresponde as k hierarquias construídas para c_i. O valor de k é sempre menor que m, uma vez que alguns segmentos de $\mathcal{S}_\mathcal{D}$ são punidos.

Para o caso onde c_i é obtida a partir da consulta $q_2 = (Conferência = \text{VLDB}, Keyword = \text{"OLAP textual "})$, apresentando pelo menos um filtro sobre a dimensão textual, DTCubing constrói uma hierarquia de tópicos para cada $[\mathcal{S}_i] \in \mathcal{S}_\mathcal{D}$ sem nenhuma punição. A diferença é que apenas os segmentos que possuem os termos "*OLAP*" e "*textual*" são utilizados para a construção das hierarquias. Observe que se um *parágrafo* possui a ocorrência de um determinado termo, não significa que todas as suas sentenças possuem a ocorrência do mesmo termo. Neste caso, o conjunto de hierarquias $\mathcal{H}_{c_i} = \{\mathcal{H}_{\mathcal{S}_1}, \ldots, \mathcal{H}_{\mathcal{S}_m}\}$ é construído para c_i.

4. ALGORITMOS

A abordagem DTCubing é particionada em quatro etapas: *Indexação*, *Filtragem*, *Construção de Hierarquias* e *Ranking*. A solução computa cubos de intervalo (*range cubes*), ou seja, cubos de dados compostos de células que representam agregações totais ou parciais. Assim, o valor ALL tradicional existe, mas também vários outros valores ALL, considerando filtros como *entre*, *similar*, *menor que* e muitos outros. Sendo assim, o comportamento exponencial em termos de tempo e consumo de memória torna impraticável as estratégias de cubos de intervalo completos, como explicado em [16, 24]. Um protótipo web da solução DTCubing está disponível online [3].

4.1 Indexação

Uma coleção de documentos ***DOC*** é definida como um conjunto de tuplas, onde cada tupla t é formalizada como $t = (TID, D_1, D_2, \ldots, D_n, \{doc_1, doc_2, \ldots, doc_{n'}\})$, onde n é igual ao número de dimensões estruturadas e n' é o número de documentos associados com a tupla t. Cada dimensão estruturada está na forma $D_j = \{at_{1,j} + at_{2,j} + \cdots + at_{k,j}\}$, onde k é a cardinalidade da dimensão e $at_{i,j}$ corresponde ao i-*ésimo* atributo da dimensão j. O símbolo $+$ corresponde ao operador lógico *OU*. TID corresponde a um identificador único, garantindo que não exista outra tupla igual em ***DOC***.

A coleção ***DOC*** é utilizada como entrada para a etapa de indexação, a qual produz como saída um cubo de dados na forma $dtc = (\ldots\left(iT_{at_{1,j}}, iT_{at_{2,j}}, \ldots, iT_{at_{n,j}}\right)\ldots T, R)$. Cada elemento interno $(iT_{at_{1,j}}, iT_{at_{2,j}}, \ldots, iT_{at_{n,j}})$ corresponde ao conjunto de tuplas invertidas de uma dimensão específica. Cada $iT_{at_{1,j}} = (at_{i,j}, TID_1, \ldots, TID_p)$ representa a lista de tuplas invertidas do atributo $at_{i,j}$, armazenando um conjunto de identificadores de tuplas $(TID_1, \ldots, TID_p)$, representando as p ocorrências de $at_{i,j}$ em ***DOC***. T representa a dimensão textual do cubo e R um conjunto de rankings construídos a partir dos segmentos de texto considerados.

O Algoritmo 1 apresenta a rotina da etapa de *Indexação*. A variável *sortedC* (linha 1) armazena a cardinalidade de cada dimensão, sendo útil para otimizar o cálculo das agregações nas etapas subsequentes. A variável *invertedT* (linha 2) armazena todos os atributos de ***DOC***, assim como suas listas de TIDs. Além dos atributos das dimensões estruturadas, os termos dos documentos de cada tupla também são indexados e armazenados em *invertedT*. Considerar os termos dos documentos como atributos, armazenando-os individualmente em um índice invertido, facilita a aplicação de filtros como *diferente*, *contém* ou *similar* na dimensão textual.

Com a variável *invertedT* é possível obter as relações "*atributo*"-"tupla" e "*termo*"-"*tupla*". No entanto, para a etapa de construção das hierarquias é necessário obter de forma eficiente os segmentos de texto pertencentes aos documentos associados com cada tupla. Para isto, a variável $\mathcal{S}$ (linha 3) armazena os segmentos de texto de cada documento pertencente a ***DOC***, estabelecendo a relação "*documento*"-"*segmentos*".

Para cada tupla de ***DOC***, o algoritmo inicialmente indexa as dimensões estruturadas (linhas 7-9). Caso um atributo at_i tenha sido lido pela primeira vez, uma nova entrada será criada na variável *invertedT*. Caso contrário, sua lista de TIDs é atualizada com uma nova ocorrência. Ao chegar na dimensão textual, o algoritmo remove as *stopWords* de cada documento (linha 11), separando seus segmentos de texto (linha12) e adicionando-os à variável $\mathcal{S}$ (linha 13). Em sequência, o algoritmo indexa todos os termos de cada documento, assim como feito para a dimensão estruturada (linhas 14-15).

Algoritmo 1: Algoritmo de Indexação

```
   Indexing(DOC)
1. int[]sortedC;
2. Map<att, Set<TID>>[] invertedT;
3. Map<docID, segments>[] S;
4. while(DOC has tuples){
5.   i=1;
6.   tuple t = DOC.tuple;
7.   while(t has dimensions)
8.     if(t.dimension.isStructured())
9.       atualiza ou cria nova entrada
         em invertedT para cada atributo;
10.      else
11.        d = removeStopWords(t.getDoc());
12.        segments = splitSegments(d);
13.        S.put(i, segments); i++;
14.        for each term in d
15.          atualiza ou cria nova entrada
             em invertedT para cada termo;
16. atualiza sortedC para manter dimensões
    de acordo com suas cardinalidaddes;
17. return dtC;
```

4.2 Filtragem

A etapa de filtragem em DTCubing abrange a aplicação de filtros pontuais, filtros de intervalo e filtros combinatoriais, detalhados em [12, 16, 24]. Nesta etapa, o usuário seleciona as células que serão adotadas para a construção das hierarquias de tópicos. Um filtro pontual possui apenas um operador de igualdade. Um filtro de intervalo pode ser classificado em: *maior que*, *menor que*, *entre*, *diferente*, *contém*, *alguns* e *similar*. Todos os filtros de intervalo incluem o atributo ALL em seus resultados. Por fim, um filtro combinatorial possui apenas um operador, que reúne todos os valores de uma dimensão mais ALL, resultando em um sub-cubo. Como resultado da etapa de filtragem, obtém-se uma ou mais células do cubo.

O Algoritmo 2 para a etapa de *Filtragem* possui como entrada o cubo parcial *dtc* obtido previamente na etapa de *Indexação,* e um

[3] http://www.dtcubing.programo.com.br

conjunto $Fil = \{f_1, f_2, \ldots, f_n\}$ de filtros definidos pelo usuário. Cada filtro f_i é aplicado a uma dimensão específica do cubo, sendo assim, o algoritmo primeiramente obtém os valores de atributos de cada dimensão, armazenando-os na variável $attValues$ (linhas 3-5).

Após definir os valores de atributo de cada dimensão, o algoritmo produz todas as possíveis agregações com os valores contidos em $attValues$ (linha 6). A estratégia de agregação implementada é *bottom-up*, começando com 1D cuboides para produzir os 2D cuboides, então adota os 2D cuboides pra produzir 3D cuboides e assim por diante. É importante ressaltar que o processo de agregação utiliza memória externa e pode gerar tuplas vazias, não existentes na coleção de documentos.

Até este passo as agregações foram processadas e armazenadas em disco. Para cada valor de atributo de uma tupla, sua lista de TIDs é recuperada e ao final é realizada a interseção dos TIDs de cada valor de atributo, com os TIDs que representam o resultado parcial da tupla. Se uma interseção parcial é vazia, o procedimento interrompe sua execução, evitando interseções desnecessárias. Uma política de cache é utilizada para armazenar os resultados parciais das interseções que estão sendo executadas.

Algoritmo 2: Algoritmo de Filtragem

```
   Filtering(dtC, Fill)
1. Set<attribute> [] attValues;
2. File file;
3. for each f_i in Fill
4.    atts = attributeValues(dtc, f_i);
5.    attValues.add(atts);
6. generateAggregations(attValues, file);
7. resultCells = parallelInters(file, dtc);
8. return resultCells;
```

4.3 Construção das Hierarquias

A etapa de *Construção das Hierarquias* é iniciada a partir do momento em que o usuário seleciona uma das células retornadas pela etapa de *Filtragem*. Os conjuntos de segmentos de texto agregados na célula selecionada são utilizados para a construção de várias hierarquias, respeitando as restrições apresentadas na Seção 3.3. Assim como descrito na Seção 3.1, cada hierarquia $\mathcal{H}_{\mathcal{S}_j}$ é organizada como uma árvore, onde cada nó é representado por uma distribuição θ_j de tópicos. Além disso, cada hierarquia armazena uma matriz de probabilidades $\mathcal{P}$, que apresenta a probabilidade de cada tópico t_i para cada distribuição θ_j. Portanto, a matriz $\mathcal{P}$ representa a probabilidade de cada tópico pertencer a cada nó da árvore hierárquica.

O Algoritmo 3 representa a criação das hierarquias de tópicos em DTCubing. O algoritmo recebe como entrada a célula c_i contendo uma lista de TIDs, a variável $\mathcal{S}$ contendo o mapeamento de cada documento para seus conjuntos de segmentos de texto e a variável $textFilter$, que assume o valor **TRUE** se c_i foi obtida por pelo menos um filtro na dimensão textual e **FALSE** caso contrário. $Topic_Hierarchies$ (linha1) é o vetor responsável por armazenar o conjunto de hierarquias de tópicos construídas para c_i.

A função $getSegments$ (linha 2) armazena em $\mathcal{S}_D$ todos os segmentos de texto relacionados a c_i. Cada posição de $\mathcal{S}_d$ armazena diferentes agregações de segmentos de texto, ou seja, $\mathcal{S}_d$ [0] armazena os *títulos* agregados em c_i, $\mathcal{S}_d$ [1] armazena os *resumos* agregados em c_i, e assim por diante. Se o valor da variável $textFilter$ for igual a **FALSE**, alguns segmentos são punidos, assim como explicado na Seção 3.3.

Algoritmo 3: Algoritmo de construção das Hierarquias de Tópicos

```
Hierarchy_Generation(c_i, S, textFilter){
1. Topic_Hierarchies[];
2. S_D = getSegments(c_i, S,textFilter);
3. i=0;
4. for each [S_i] in S_D
5.   if(textFilter){
6.     H_Si = createHierarchy([S_i]);
7.     Topic_Hierarchies[i] = H_Si;
8.     i++;
9.  c_i.setHierarchies(Topic_Hierarchies);
```

A função $createHierarchy$ (linha 6) é responsável por realizar a chamada do método CATYH [3], responsável pela construção das hierarquias, onde é possível obter a matriz de probabilidade $\mathcal{P}$ associada com cada $\mathcal{H}_{\mathcal{S}_i}$. Na linha 7, cada hierarquia $\mathcal{H}_{\mathcal{S}_i}$ é adicionada ao vetor $Topic_Hierarchies$, que ao final do processo, contém as hierarquias construídas a partir de todos os conjuntos de segmentos de texto armazenados em $\mathcal{S}_d$. Finalmente, o conjunto de hierarquias é adicionado à célula c_i (linha 9), permitindo que o usuário navegue através de diversas perspectivas diferentes sobre um mesmo conjunto de dados, através dos níveis apresentados pelas diversas hierarquias de tópicos.

4.4 Ranking

A etapa de construção dos rankings de segmentos se inicia a partir do momento em que o usuário seleciona um nó particular de uma das hierarquias de tópicos construídas na etapa anterior.
O resultado dessa etapa oferece ao usuário um ranking dos segmentos de texto mais relevantes à distribuição de tópicos (nó) selecionada. Assim como em [23], DTCubing adota uma distribuição de probabilidades a fim de obter uma medida de cobertura de cada segmento de texto, para cada distribuição de tópicos apresentada pelos nós das hierarquias. Essa distribuição de probabilidades é obtida através da matriz $\mathcal{P}$, agregada em cada hierarquia $\mathcal{H}_i$, onde cada linha corresponde a um tópico e cada coluna corresponde a uma distribuição de tópicos (nó) de $\mathcal{H}_i$. Cada posição (i, j) de $\mathcal{P}$ corresponde à probabilidade do tópico t_i pertencer à distribuição θ_j. Essa probabilidade pode ser formalizada como $p(t_i|\theta_j)$.

Todo tópico pertence a um segmento de texto e, portanto, é possível computar uma medida de cobertura que determina a probabilidade de um segmento s_k pertencer a uma dada distribuição θ_j, apenas somando os valores de probabilidades dos tópicos encontrados em s_k. Sendo assim, seja $p(\theta_j|s_k) = \sum_{t_i \in s_k} p(t_i|\theta_j)$ a probabilidade de um segmento s_k cobrir a distribuição θ_j, representada pela soma das probabilidades de cada $t_i \in s_k$, de acordo com a distribuição θ_j. Essa estratégia é apresentada em Topic Cube [23].

O Algoritmo 4 representa o processo de construção dos rankings em DTCubing, recebendo como parâmetro a distribuição de tópicos θ_j pertencente ao nó escolhido pelo usuário, o conjunto $[\mathcal{S}i]$ de segmentos agregados na célula escolhida na etapa de *Filtragem* e a matriz de probabilidades $\mathcal{P}$. Nas linhas 2-4 é computada a cobertura de cada segmento $s_k \in [\mathcal{S}i]$ a partir de θ_j. Ao final dessa etapa, todos os valores de cobertura são ordenados, apresentando primeiro aqueles segmentos que se fizeram mais relevantes à θ_j, ou seja, os segmentos que obtiveram os maiores valores de cobertura.

Algoritmo 4: Algoritmo de Ranking de Segmentos

```
Ranking(θ_{i,j}, [S_i], P){
1.  Map<SegmentID, cover> ranking;
2.  for each s_k in [S_i]{
3.     cover = Covering(θ_{i,j}, s_k, P);
4.     ranking.put(s_k.getID, cover);
5.  sort(ranking);
```

5. VALIDAÇÃO DAS HIPÓTESES

Esta seção apresenta a validação das hipóteses apresentadas no escopo deste trabalho. A base de dados utilizada na fase experimental foi obtida em [18], contendo um conjunto de artigos científicos coletados da DBLP. Cada artigo contém os campos: *título*, *autor*, *ano*, *conferência* e *resumo*. Os segmentos de texto *título* e *resumo* foram indexados como dimensões textuais, enquanto os outros campos foram indexados como dimensões estruturadas. Foram consideradas as 30 conferências com o maior número de artigos, totalizando 80.000 artigos científicos.

Cada hierarquia de tópicos construída seguiu uma mesma topologia de árvore, contendo 13 nós, organizados em três níveis hierárquicos, similar ao trabalho [23]. O primeiro nível agrega apenas o nó raiz, o qual possui 4 filhos, que constituem o segundo nível. Cada filho do nó raiz possui outros dois filhos, totalizando 8 nós no terceiro e último nível. Cada nó de uma hierarquia possui um total de 10 tópicos, menos o nó raiz, que representa a união de todos os tópicos contidos na árvore.

5.1 Múltiplas Hierarquias de Tópicos

O objetivo desta seção é comprovar que as múltiplas hierarquias dinâmicas construídas em DTCubing apresentam tópicos diferentes e mais específicos quando comparados à hierarquia global proposta em [19]. A hierarquia de tópicos construída na abordagem EventCube é aqui formalizada como Hierarquia Global, ou HG para abreviar. As métricas *cosseno* [7] e *entropia* [1] foram utilizadas para garantir a exclusividade e especificidade dos tópicos listados pelas hierarquias comparadas.

A métrica *cosseno* quantifica a similaridade entre dois conjuntos, sendo definida no intervalo [0,1]. Quanto mais próximo de zero, menor a similaridade entre os conjuntos comparados. Seja $\overrightarrow{n_a}$ e $\overrightarrow{n_b}$ dois nós representados pelos seus vetores de termos. A similaridade cosseno entre eles é dada de acordo com a equação 1:

$$SIM_c(\overrightarrow{n_a}, \overrightarrow{n_b}) = \frac{\overrightarrow{n_a} \cdot \overrightarrow{n_b}}{|\overrightarrow{n_a}| \times |\overrightarrow{n_b}|} \quad (1)$$

onde $\overrightarrow{n_a}$ e $\overrightarrow{n_b}$ são vetores multidimensionais e cada dimensão representa um termo com o seu peso (não negativo) no documento. Cada nó n_i de HG foi comparado com todos os nós das hierarquias DTCubing, produzindo um conjunto de matrizes de similaridade. Para melhor visualização e entendimento, cada matriz de similaridade é apresentada na forma de um mapa de calor.

Assumimos que a hierarquia apresentada por EventCube pode ser construída a partir de qualquer segmento de texto para que a comparação possa ser justa. Seja HG_T e HG_R as hierarquias gerais construídas a partir dos segmentos de texto *título* e *resumo,* respectivamente. A Figura 2 ilustra a comparação das hierarquias de DTCubing que apresentaram as maiores e as menores similaridades se comparadas a HG_T e HG_R . Utilizando o segmento de texto *título* como base para a construção das hierarquias, a hierarquia construída para a conferência VLDB se mostrou a menos similar à HG_T, enquanto a hierarquia construída para a conferência IEICE foi a que obteve o maior valor de similaridade. Já para o segmento de texto *resumo*, as hierarquias construídas para as conferências AAAI e IEICE foram as que apresentaram a menor e a maior similaridade se comparadas à HG_R, respectivamente.

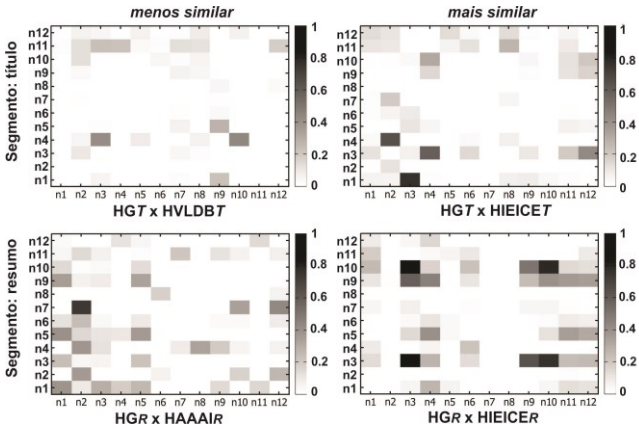

Figura 1. Matrizes de similaridade cosseno das hierarquias DTCubing comparada a HG_T e HG_R.

A maioria dos valores de cosseno ilustrados na Figura 1 estão próximos de zero, resultando em mapas de calor com células predominantemente brancas. Mesmo as hierarquias que se mostraram mais similares à HG_T e HG_R (construídas a partir dos *títulos* e dos *resumos* dos documentos pertencentes à conferência IEICE) possuem valores de cosseno bem baixos. O maior valor de similaridade obtido utilizando a métrica *cosseno* é igual a 0,1445, demonstrando que DTCubing apresenta hierarquias com tópicos bem distintos daqueles apresentados pela literatura.

A métrica *entropia* [1] foi utilizada a fim de obter comprovações matemáticas de que, além de produzir tópicos diferentes, as hierarquias construídas em DTCubing também apresentam tópicos mais específicos ao contexto da consulta do usuário. Diferentemente do *cosseno*, a métrica *entropia* não é uma medida de similaridade, sendo aplicada individualmente, a fim de obter o nível de coesividade de um conjunto. Quanto menor o valor de *entropia*, mais coeso o conjunto analisado, tendendo a ser mais focado em um único tópico [1].

Seja $\mathcal{H}_j$ uma hierarquia de tópicos construída em DTCubing. O valor de entropia é computado para a concatenação dos tópicos de todos os 12 nós de $\mathcal{H}_j$. Cada tópico é composto de um ou mais termos, sendo assim, seja $D = \{t_1, t_2, \dots, t_n\}$ o conjunto de todos os termos que compõem todos os tópicos listados em $\mathcal{H}_j$. O valor da entropia é calculado de acordo com a Equação 2:

$$-\sum_{t \in D} p_D(t) \log p_D(t) \quad (2)$$

onde a probabilidade de um termo t_i é igual a $p_D(t_i) = \frac{tf_{t_i,D}}{\sum_{t_j \in D} tf_{t_j,D}}$.

O gráfico da Figura 2 ilustra a diferença dos valores de entropia obtidos por HG_T e por diferentes hierarquias de DTCubing. Todos os valores apresentados foram computados para hierarquias construídas a partir do segmento *título*. O primeiro valor de entropia, destacado em laranja, corresponde ao valor calculado para

os tópicos de HG_T. O segundo valor, destacado em azul, corresponde à entropia calculada para a hierarquia construída a partir dos *títulos* dos documentos agregados na célula obtida pela aplicação do filtro $(Conference = ICC)$. Os demais valores, destacados em cinza, referem-se aos valores de entropia computados para as hierarquias construídas para as células obtidas pelos filtros $(Conferência = ICC,\ Ano = 2007)$, $(Conferência = ICC,\ Ano = 2008)$, …, $(Conferência = ICC,\ Ano = 2015)$. Observe que, à medida em que o usuário vai especializando sua consulta, os valores de entropia vão se tornando menores. Deste modo é possível provar que além de apresentar tópicos exclusivos, as hierarquias construídas em DTCubing apresentam tópicos mais específicos ao contexto da consulta realizada pelo usuário. O mesmo comportamento da métrica *entropia* foi observado para os demais segmentos utilizados em DTCubing.

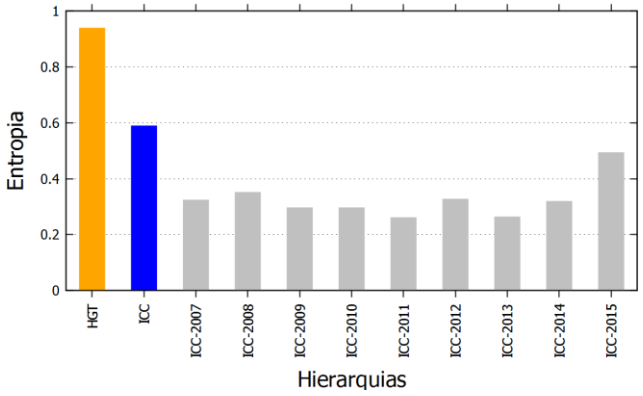

Figura 2. Valores de entropia obtidos para HG_T e para diferentes hierarquias DTCubing.

5.2 Rankings de Segmentos

O objetivo desta seção é comparar a diferença entre os rankings de segmentos mais relevantes obtidos pelas hierarquias DTCubing e por HG (literatura). Todos os rankings são comparados através do cálculo da métrica *jaccard*. O coeficiente de *jaccard* é uma métrica de similaridade calculada pela interseção de dois conjuntos, dividido pela união dos mesmos conjuntos [7]. A métrica é calculada de acordo com a Equação 3:

$$SIM_j(A,B) = \frac{|A \cap B|}{|A \cup B|} \qquad (3)$$

Assim como o *cosseno*, a métrica *jaccard* também está definida no intervalo [0,1]. Quanto mais próximo de 1 maior será a similaridade entre os rankings comparados, resultando em uma maior redundância. Seja c_i uma célula DTCubing e $\mathcal{H}_{c_iT}$ e $\mathcal{H}_{c_iR}$ as hierarquias construídas a partir dos *títulos* e dos *resumos* agregados em c_i. O processo de ranqueamento de cada segmento de texto depende da distribuição de tópicos apresentada por cada hierarquia. O objetivo é comparar a diferença ao se ranquear os segmentos de c_i a partir da distribuição de tópicos apresentada por $\mathcal{H}_{c_iT}$ e $\mathcal{H}_{c_iR}$, e ao se ranquear os mesmos segmentos a partir das distribuições de tópicos apresentada por HG_T e HG_R. A comparação é feita apenas para hierarquias construídas a partir do mesmo segmento. Além disso, verifica-se também o impacto do tamanho de k em DTCubing, onde k corresponde ao número de segmentos mais relevantes retornados em cada ranking.

A Figura 3 ilustra um conjunto de matrizes jaccard, obtidas ao se ranquear os títulos dos documentos pertencentes à conferência ACM a partir da distribuição de tópicos apresentada por $\mathcal{H}_{ACM_T}$, e ao se ranquear o mesmo conjunto de segmentos a partir da distribuição de tópicos apresentada por HG_T. Cada matriz de similaridade corresponde a um valor diferente de k, variando entre 10, 20, 40, 60, 80 e 100. $\mathcal{H}_{ACM_T}$ corresponde à hierarquia construída a partir dos títulos da conferência ACM, sendo esta escolhida por apresentar os maiores valores de similaridade *jaccard*.

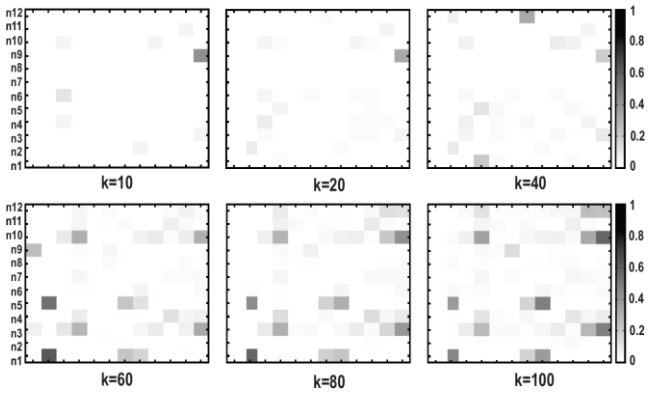

Figura 3. Valores de jaccard obtidos ao se comparar os rankings de HG_T e $\mathcal{H}_{ACM_T}$ para diferentes valores de k.

Observe que mesmo apresentando os maiores valores obtidos pela métrica *jaccard*, os rankings obtidos em $\mathcal{H}_{ACM_T}$ ainda se diferem bastante dos rankings obtidos por HG_T. À medida em que o valor de k aumenta, os rankings tendem a conter cada vez mais segmentos iguais, tornando-se cada vez mais similares. No entanto, mesmo para $k = 100$, é possível observar que a maioria das células do mapa de calor apresentado na Figura 3 possui uma tonalidade bem clara, tendendo a um valor de jaccard igual a zero. Mais precisamente, o maior valor de jaccard obtido dentre as comparações foi igual a 0,57.

Este resultado comprova que os documentos considerados mais relevantes por DTCubing são diferentes daqueles considerados mais relevantes pela literatura. Logo, fica evidente que calcular a medida de cobertura de cada segmento/documento sempre a partir da mesma distribuição de tópicos pode não ser a melhor estratégia, uma vez que os tópicos apresentados por HG são muito genéricos, fazendo com que documentos importantes percam relevância diante do contexto apresentado por cada célula.

6. CONCLUSÃO

Neste trabalho é apresentada uma nova abordagem OLAP textual denominada DTCubing. O objetivo é propor uma criação automática e dinâmica de múltiplas hierarquias de tópicos por cada célula do cubo. Como consequência das múltiplas hierarquias, diversos rankings de segmentos são apresentados aprimorando o resultado das consultas multidimensionais presentes na literatura.

Os experimentos comprovaram a utilidade de DTCubing mediante diversos cenários, demonstrando que as múltiplas hierarquias dinâmicas apresentam tópicos exclusivos e mais relevantes ao contexto de cada célula. A apresentação de diversos rankings oferece uma maior variedade de resultados ao usuário, priorizando segmentos de texto que são mais relevantes aos tópicos apresentados pelas hierarquias.

Como trabalhos futuros sugere-se a validação das hipóteses através de outros conjuntos de dados, com o objetivo de reafirmar os

resultados obtidos na fase de experimentação. Além disso, se faz necessária a avaliação de desempenho das etapas de *Indexação*, *Filtragem*, *Construção das Hierarquias* e *Ranking*, apresentando a complexidade de cada algoritmo diante do conjunto de dados utilizado e do tipo de consulta realizada pelo usuário.

7. REFERÊNCIAS

[1] Bendersky, M., Croft, W.B. and Diao, Y. 2011. Quality-biased Ranking of Web Documents. *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining (New York, NY, USA, 2011)*, 95–104.

[2] Bouakkaz, Mustapha, Sabile Loudcher, and Youcef Ouinten. "OLAP textual aggregation approach using the Google similarity distance." *International Journal of Business Intelligence and Data Mining* 11.1 (2016): 31-48.

[3] Wang, Chi, et al. A phrase mining framework for recursive construction of a topical hierarchy. *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2013.

[4] Cuzzocrea, A., De Maio, C., Fenza, G., Loia, V. and Parente, M. 2016. OLAP Analysis of Multidimensional Tweet Streams for Supporting Advanced Analytics. *Proceedings of the 31st Annual ACM Symposium on Applied Computing (New York, NY, USA, 2016),* 992–999.

[5] Ding, B., Zhao, B., Lin, C.X., Han, J., Zhai, C.X., Srivastava, A. and Oza, N.C. 2011. Efficient Keyword-Based Search for Top-K Cells in Text Cube. *IEEE Transactions on Knowledge and Data Engineering*. 23, 12 (Dec. 2011), 1795–1810.

[6] Gray, Jim, et al. "Data cube: A relational aggregation operator generalizing group-by, cross-tab, and sub-totals." *Data mining and knowledge discovery* 1.1 (1997): 29-53.

[7] Huang, Anna. "Similarity measures for text document clustering." *Proceedings of the sixth new zealand computer science research student conference (NZCSRSC2008), Christchurch, New Zealand*. 2008.

[8] Janet, B. and Reddy, A.V. 2010. Cube Index: A Text Index Model for Retrieval and Mining. *International Journal of Computer Applications.* 1, 9 (Feb. 2010), 84–89.

[9] Janet, B. and Reddy, A.V. 2011. Cube Index for Unstructured Text Analysis and Mining. *Proceedings of the 2011 International Conference on Communication, Computing & Security* (New York, NY, USA, 2011), 397–402.

[10] Lauw, H.W., Lim, E.-P. and Pang, H. 2007. TUBE (Text-cUBE) for Discovering Documentary Evidence of Associations Among Entities. *Proceedings of the 2007 ACM Symposium on Applied Computing* (New York, NY, USA, 2007), 824–828.

[11] Lee, S., Kim, N. and Kim, J. 2014. A Multi-dimensional Analysis and Data Cube for Unstructured Text and Social Media. *2014 IEEE Fourth International Conference on Big Data and Cloud Computing (BdCloud)* (Dec. 2014), 761–764.

[12] Li, X., Han, J. and Gonzalez, H. 2004. High-dimensional OLAP: A Minimal Cubing Approach. *Proceedings of the Thirtieth International Conference on Very Large Data Bases - Volume 30* (Toronto, Canada, 2004), 528–539.

[13] Lin, C.X., Ding, B., Han, J., Zhu, F. and Zhao, B. 2008. Text Cube: Computing IR Measures for Multidimensional Text Database Analysis. *Eighth IEEE International Conference on Data Mining,* 2008. ICDM '08 (Dec. 2008), 905–910.

[14] Liu, X., Tang, K., Hancock, J., Han, J., Song, M., Xu, R., Manikonda, V. and Pokorny, B. 2012. SocialCube: A Text Cube Framework for Analyzing Social Media Data. *2012 International Conference on Social Informatics (SocialInformatics)* (Dec. 2012), 252–259.

[15] Oukid, L., Asfari, O., Bentayeb, F., Benblidia, N. and Boussaid, O. 2013. CXT-cube: Contextual Text Cube Model and Aggregation Operator for Text OLAP. *Proceedings of the Sixteenth International Workshop on Data Warehousing and OLAP* (New York, NY, USA, 2013), 27–32.

[16] Silva, R.R., Lima, J. de C. and Hirata, C.M. 2013. qCube: Efficient integration of range query operators over a high dimension data cube. *Journal of Information and Data Management. 4, 3* (Sep. 2013), 469.

[17] Sun, Y., Yu, Y. and Han, J. 2009. Ranking-based Clustering of Heterogeneous Information Networks with Star Network Schema. *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (New York, NY, USA, 2009), 797–806.

[18] Tang, J., Zhang, J., Yao, L., Li, J., Zhang, L. and Su, Z. 2008. ArnetMiner: Extraction and Mining of Academic Social Networks. *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (New York, NY, USA, 2008), 990–998.

[19] Tao, F. et al. 2013. EventCube: Multi-dimensional Search and Mining of Structured and Text Data. *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (New York, NY, USA, 2013), 1494–1497.

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

[21] Yu, Y., Lin, C.X., Sun, Y., Chen, C., Han, J., Liao, B., Wu, T., Zhai, C., Zhang, D. and Zhao, B. 2009. iNextCube: Information Network-enhanced Text Cube. *Proc. VLDB Endow.* 2, 2 (Aug. 2009), 1622–1625.

[22] Zhang, Duo, ChengXiang Zhai, and Jiawei Han. "MiTexCube: MicroTextCluster Cube for Online Analysis of Text Cells." *CIDU*. 2011.

[23] Zhang, D., Zhai, C. and Han, J. 2009. Topic cube: Topic modeling for olap on multidimensional text databases. *In Proc. of the SIAM International Conference on Data Mining* (SDM (2009), 1123–1134.

[24] Silva, R. R., Hirata, C. M., & de Castro Lima, J. (2015, April). A Hybrid Memory Data Cube Approach for High Dimension Relations. *In ICEIS (1)* (pp. 139-149).