# Chapter

# Text-mining Similarity Approximation Operators for Opinion Mining in BI tools

PAWEŁ KAPŁANSKI, NINA RIZUN
*Gdansk University of Technology*
*Department of Applied Informatics in Management*
*Faculty of Management and Economics*
*{pawel.kaplanski, nina.rizun}@zie.pg.gda.pl*

YURII TARANENKO
*Alfred Nobel University, Dnipropetrovs'k*
*Department of Applied Linguistics and Methods*
*of Teaching Foreign Languages*
*taranen@rambler.ru*

ALESSANDRO SEGANTI
*Cognitum, Warsaw, Poland*
*a.seganti@cognitum.eu*

### Abstract

*The concept of the Text-mining Similarity Approximation Operators for Opinion Mining as extensions to Natural Language Interface Database is defined. The new operators: "keywords of" dimension; subsetting operator "about C is q"; aggregation operator "by similar C" are proposed. These operators are based on the Latent Semantic Analysis and Social Network Analysis.*

## 1. INTRODUCTION

Large number of opinions is easily accessible nowadays. It is desirable to understand their properties as they potentially contain valuable business information. Opinion mining (including sentiment analysis) tries to extract these valuable in-formation using complex Natural Language Processing (NLP) algorithms, which can be divided into the following groups: linguistic analysis and statistical analysis.

Nowadays, it started to be well understood that opinion mining is highly domain dependent making the number of opinion mining algorithms to grow year by year, however they are usually based on an intensive use of combination of simpler tasks. Given the possibility to combine these simple text semantic analysis tasks in intuitive way will make the analyst able to define her/his own opinion mining metrics.

In this paper we introduce following operators for Natural Language Interface Database (NLIDB):

1) *Dimension definition "keywords"* – based on TF/IDF transformation. Example query: "keywords by similar opinion".
2) *Subsetting operator "about C is q"* – based on Latent Semantic Analysis (LSA). Example query: "count shops by country where opinion about coffee is good on a map".
3) *Aggregation operator "by similar C"* – based on additional Social Network Analysis (SNA) Modularity Optimization. Example query: "count clients by similar opinion" These operators are implemented in OLAP cube model and are currently being implemented in BI tool called Ask.

These operators are implemented in OLAP cube model and are currently being implemented in BI tool called Ask Data Anything (ADA) introduced and evaluated in [18]. The multidimensional OLAP cube approach, is widely used and adopted. It provides decision-makers with online access to analytical capabilities based on the idea of dimensions, deals with dimensions and measurements. Each subnode of the OLAP cube contains the accumulated value of measurement calculated using the original data that fits into it (see Figure 1) and there is the need for a flexible and straightforward way of defining dimensions and the relationship between the dimensions. This can be realized with use of ontologies [6].

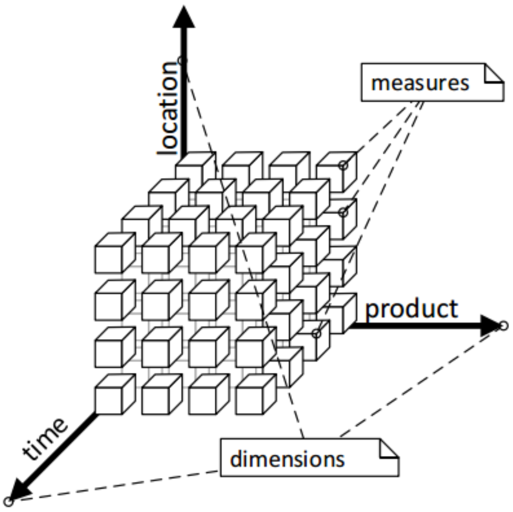


Fig. 1. OLAP data-cube

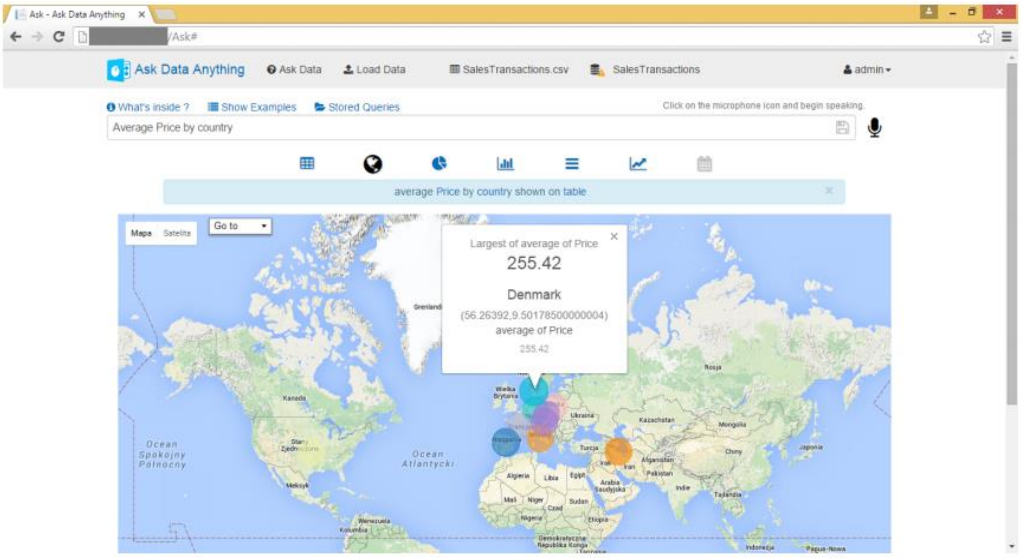
This paper has the following structure: in Section 2 authors proposed the general description of the Natural Query Language syntax and extended specification the features of the language implemented in the ADA. This section contains the typical examples of the main operator's usage for implementing the SQL queries. In Section 3 authors present *concept* of the four-step Dimensional Model of Client Text-Opinions Space to implement Text-mining Similarity Approximation Operators for Opinion Mining. This concept is based on the hypothesis of the possibility of using methods Latent Semantic Analysis and methods of the Social Network Analysis theory for the text-messages clustering task solving. In Section 4, the authors present the experimental results of the founded *concept* usage to perform queries that access the database of the text-opinions collection of the Starbucks coffee shop customer's network. In the Conclusion presents a synthesis of the research results and the main directions of further development of the presented by the authors Natural Language Interface Database operators concept.

## 2. NATURAL QUERY LANGUAGE

OLAP Cubes allow one to execute a query in a specific query language like e.g.: MDX [11]. To create a query it is required to have both: the ability to use language, and knowledge about the structure of the underlying data, and as a consequence, often data-driven decision-making cannot be used directly by the decision-maker. In other words: it is desired by the decision-maker to have the ability to examine the data in a query-result loop, where the query is tailored within an interactive process that does not require any large prior learning and preparation. This way of querying data is supported by Natural Query Language (NQL).

The typical architecture of a NQL oriented application consists of three components: (1) an NQL-based user query interface that is also responsible for the transformation of a natural language query into a formal, machine-readable database query, (2) an underlying database system and (3) a textual or graphical reporting component that presents the results of database computations. We call such systems Natural Language Interface Databases (NLIDB).

ADA is a web-browser based NLIDB. The ADA UI allows a NQL query to be entered and executed with the support of a predictive editor. The result-set of the query execution is presented on a wide range of reports including tables, charts and maps (Figure 2).



Click on the microphone icon and begin speaking.

Average Price by country

average Price by country shown on table

Largest of average of Price
255.42
Denmark
(56.26392,9.50178500000004)
average of Price
255.42

Fig. 2. ADA screen-shot from query execution result-set presented on a map

ADA NQL queries should follow EBNF grammar presented in Figure 3 and if they do not stick to the grammar, parser first tries to tailor them accordingly. Usually a query starts with an operation (1) specification (sum, average,...) followed by (possibly more than one) dimension (2) specification. The dimension specification(s) is (are) the only required grammatical part(s), all others are optional. The next part of a query defines the subsetting (3) of the data represented by the dimension, by which it is possible to filter the results. The fourth part is the aggregation (4) which allows data to be grouped in subsets. Finally, it is possible to specify the expected visualization (5) type (that can be changed later-on).

```
<query> := [<operation>] <dimension> {'and' <dimension> | <subsetting> | <aggregation>} [<output>]
<operation> := 'sum' | 'average' | 'count' | 'maximum' | 'minimum'
<subsetting> := 'where' <dimension> <compare operator> <value>
    | 'in' <ontology element> [ ( <column> ) ]
    |  in <data value>
    | 'in' <date>
    | 'from' <date> ['to ' <date>]
<aggregation> := 'by' (<dimension>|<location>|<time duration >|<ontology element>)
<output> := 'on' ['a'] ('table' | 'histogram' | 'stacked-bar' | 'map' | 'piechart' | 'line' | 'timeline )
<ontology element> := <name> ['that' 'is' <ontorion concept expression>]
<dimension> := <name>
<compare operator> := '<'|'< ='|'='|'> ='|'='|'<>'
<name> := <letter>{<digit>|<letter>|'-'}
```

Fig. 3. Simplified EBNF syntax for ADA NQL

a) *Operation*: An Operation (optional) is an action we can perform on data to get the desired information: sum, average, count, maximum and minimum.

b) *Dimension*: Every action requires at-least one Dimension specification to act on. A Dimension is assigned with a type inferred by parsing a subset of the data together with the information modeled in the supporting ontology. Currently, the type supported by the ADA CNL language are: numerical, date/time and text, for the types understood directly from the data and: location/geolocation, latitude and longitude, hierarchical (text dimension defined in the supporting ontology that can have super concepts grouping the values (e.g. american-brand for a column with brands,...) and row (dimensions that are defined in the supporting ontology and represent data from multiple columns in a single row).

Operations and types are matched in the parser to check that the query makes sense (e.g. “Sum Product” where product is a dimension with Text values inside is not allowed but “Sum Some-Row” where Some-Row is a row that contains a numerical dimension is allowed).

The query language allows also the use of:

- **ontology concepts** – names of concepts defined in the supporting ontology or complex expressions (see <ontorion concept expression> in Figure 3 and Figure 4) that are evaluated to concepts, mainly used for subsetting and grouping, described later;
- **ontology instances or literal values** (e.g. Contract 123, Rome,...) used for subsetting;

These components (as for the dimensions) are dependent on the dataset currently loaded.

c) *Subsetting*: The subsetting part of the query can be used to define the filters that the user wants to apply to his/her query. The general syntax is (Dimension, Relation, Data) whereas described before, the Relation and the Dimension are matched by the dimension type (thus Dimension > 4 is allowed only for Numerical dimensions). In this part of the query, it is also possible to use “in” constraints. After the in constraint, we expect an entity declared in the ontology (e.g. location âĂIJcontinentâĂİ, class of abstractions like “american-brand”) or the content of some column.

Complex ontology concepts have form of OCNL expressions that evaluate within OCNL grammar to concepts (see Figure 4), and are connected via <ontorion concept expression> non-terminal symbol therefore we can say that OCNL is embedded in ADA NQL (idea of embedded con-trolled languages is presented in [14]).

```
<ontorion concept expression> := ['not'] (('is'|'be'|'are') <object> | <role > <object restriction>)

<object> := ['a'|'an'] <single> | 'something' [<that>] | <instance > | 'nothing'

<role> := <small name> | ('is'|'be'|'are') <small name> 'by'

<object restriction> := <object>

        | ('nothing-but'|<comparer><count>) (<single>|<instance >|'something' <that>)

        | 'none'

        | 'itself '
```

```
<single> := <small name> [<that>] | 'thing' | 'things '
<that> := 'that' <intersection of union of ontorion concept expressions >
      | '(' 'that' <intersection of union of ontorion concept expressions > ')'

      | '(' 'that-is-one-of:' <instance > {',' <instance>} ')'
<intersection of union of concept expressions >
            := <ontorion concept expression> {('and' | 'and-or') <ontorion concept expression>}

<instance > := <big name>
<comparer> := ['at-most'|'at-least'|'less-than'|'more-than'|'different-than']
<small name> := <small letter> {<digit>|<small letter>|<big letter>|'-'}
<count> := ('no'| 'single' | 'two' | 'three' |...| 'ten') | {<digit>}+
<big name> := <big letter>{<digit>|<small letter>|<big letter>|'-'}
<digit> := '0'|1'|'2'| |'9'
<small letter> := 'a'|'b'|'c'| |'z'
<big letter> := 'A'|'B'|'C'| |'Z'
```

Fig. 4. Simplified version of EBNF Grammar for OCNL complex concept specification

d) *Example*: Let’s consider the following query: *Sum price in fish that has-size greater-than 10 and is not a frozen-product and-or is a freshwater-fish on a piechart*. This query contains the following <ontorion concept expression>: “fish that has-size greater-than 10 and is not a frozen-product and-or is a freshwater-fish” that evaluates into DL concept expression: fish∏ ((∃ haveâĹŠsize>10∏ ¬:frozenâĹŠproduct)∐ freshwaterâĹŠfish).

During reasoning process, that takes place in Ontology Management System, we obtain set of instances of the afore-mentioned complex concept expression: (Fish-1, Fish-2,...) that are then injected into final SQL query:

```
select sum ( price) from dataset
where fish in ( Fish – 1, Fish – 2 , . . . )
```

If the element is the value of a column that appears in more than one column, it is possible to point the parser to the correct column by adding its name in parenthesis.

Subsetting by date is very expressive, for example the user can write: “from year 2015 to/until year 2016”, “from July 2015 to/until September 2015”, “from 1st of July 2015 to/until 23rd of October 2015”, “from 07/01/2015 to/until 08/02/2015” or “from 07/01/2015 12:23 to/until 08/02/2015 09:22”.

e) *Aggregation*: Aggregation is the action of grouping the result using one of the Dimensions and/or entities which were defined in the ontology; the syntax for aggregation is described in Figure 3. The syntax for aggregation is âĂIJbyâĂİ together with dimension, location (i.e. continent, country), and time period (year, month, day, date). Multiple aggregations are allowed (e.g. by country and by day).

Some aggregations require operations and others do not (e.g. by day can be used with or without operations on the dimension, while by country needs an operation).

f) *Outputs*: It is possible to specify in the query language the output on which the query result should be shown. ADA currently support following types of outputs: table, histogram, stacked-bar, map, pie chart, line or timeline. After the query is parsed, the parser decides which of the outputs are allowed depending on the type of dimensions that will be returned.

## 3. Text-Mining Similarity Approximation Operators For Opinion Mining Preprocessing

Text-mining Similarity Approximation Operators for Opinion Mining are defined as extensions to previously introduced NLIDB called ADA. To implement the proposed concept of Text-mining Similarity Approximation Operators in BI tools necessary to carry out the following four Preprocessing steps of the **Dimensional Modeling of Client Text-Opinions Space** (fig. 5) [16]:

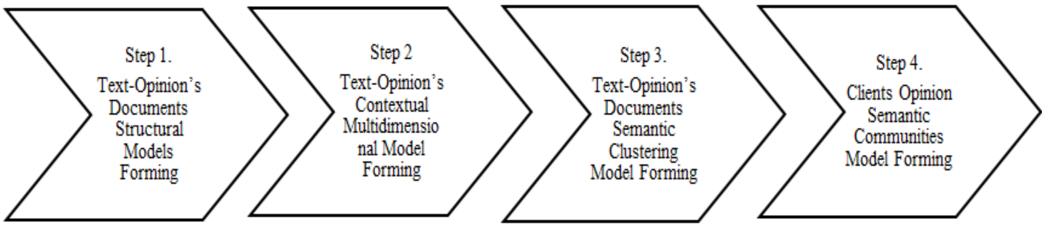

Step 1.
Text-Opinion's Documents Structural Models Forming
Step 2
Text-Opinion's Contextual Multidimensio nal Model Forming
Step 3.
Text-Opinion's Documents Semantic Clustering Model Forming
Step 4.
Clients Opinion Semantic Communities Model Forming

Fig. 5. Preprocessing Steps of the Dimensional Modeling of Client Text-Opinions Space

### 3.1. The Text-Opinion's Documents Structural Models Forming

As a basic concept of forming the **structural model of the text-opinion's document**, is the frequency $D_{w_i}$ of the words occurrence in the documents. As a mathematical apparatus of the model's formalization stage is proposed to use the *TF/IDF* transformation [7, 8, 15, 16] of the frequency matrix – the statistical measure used for evaluation of the significance of the terms in the context of the document, which is part of the list of analyzed documents:

$$D_{w_i} = TFIDF_N(w,t) = TF(w,t) \bullet IDF_N(w,\tau), \quad (1)$$

where:

– term *w* is the keyword: forms of the word (stemming results – the procedure of the words bases selection) except for the words that have no semantic load (prepositions, pronouns, etc.);

– relative frequency $TF(w,t)$ of the *w*-th term occurrence in document:

$$TF(w,t) = \frac{k(w,L_t)}{S(L_t)} \tag{2}$$

where $k(w,L_t)$ – the number of *w*-th term occurrences in the *t*-th document *L*; $S(L_t)$ – the total number of terms in the *t*-th document *L*;

– standardized value inverse frequency $IDF_N(w,\tau)$ of the w-*th* term occurrences in a set of analyzed documents *w*.

$$IDF_N(w,\tau) = \frac{\log_2 \dfrac{n}{m(w,\tau)}}{\log_2 n} = \log_n \frac{n}{m(w,\tau)} \tag{3}$$

where *n* – the total number of the documents in a set τ; $m(w,\tau)$ – the number of the documents in a set τ, which contains the term.

As a **result of the first step** of the Dimensional Modeling of Client Text-Opinions Space: for each document we have the **plane frequency model** $Fr_t = \langle \phi_t, D_t \rangle$, which is offered as a combination of the following elements (fig. 6):

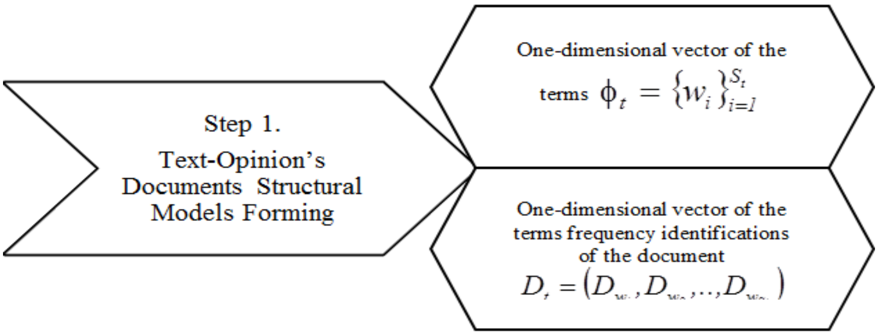

Step 1.
Text-Opinion's Documents Structural Models Forming
One-dimensional vector of the terms $\phi_t = \{w_i\}_{i=1}^{S_t}$
One-dimensional vector of the terms frequency identifications of the document $D_t = (D_{w_1}, D_{w_2}, \dots, D_{w_{S_t}})$

Fig. 6. The results of the first step of the Dimensional Modeling of Client Text-Opinions Space

## 3.2. The Text-Opinion's Contextual Multidimensional Model Forming

The next step is the process of knowledge extracting from the Client Text-Opinions Space in order to form its Context Multidimensional Model.

As a method of knowledge extraction from customer's text-opinions collection in order to create the multidimensional document's model proposes to use the tools of LSA [7, 8, 15, 16, 20]. The most common version of LSA is based on the *Singular Value Decomposition* (SVD), which allows reflecting basic structure of the different dependencies that are present in the original matrix. The mathematical basis of the method is as follows:

Building the matrix *A* of the frequency of occurrence the indexed *terms* (keywords) in the documents starts with sequential procedures of removing words, which do not have any sense load (prepositions, pronouns etc.); removing words that are found only once in the whole document; stemming (finding the word’s stem – for instance, the Porter’s rule).

Formally let *A* be the $m \times n$ term-document matrix of the document’s collection. Each column of *A* corresponds to a document. The values of the matrix elements $A[i,j]$ represent the frequency identifications $D_{w_i}$ of the term occurrence $w_i$ in the document: $A[i,j] = D_{w_i}$. The dimensions of *A*, *m* and *n*, correspond to the number of words and documents, respectively, in the collection. Observe that $B = A^T A$ is the document-document matrix. If documents *i* and *j* have *b* words in common then $B[i,j] = b$. On the other hand, $C = AA^T$ is the term-term matrix. If terms *i* and *j* occur together in *c* documents then $C[i,j] = c$. Clearly, both *B* and *C* are square and symmetric; *B* is an $m \times m$ matrix, whereas *C* is an $n \times n$ matrix.

Singular Value Decomposition (SVD) of *A* using matrices *B* and *C* is defined as $A = S\Sigma U^T$, where *S* is the matrix of the eigenvectors of *B*, *U* is the matrix of the eigenvectors of *C*, and $\Sigma$ is the diagonal matrix of the singular values obtained as square roots of the eigenvalues of *B*.

In *LSA* we ignore these small singular values and replace them by 0. Let us say that we only keep *k* singular values in $\Sigma$. Then $\Sigma$ will be all zeros except the first k entries along its diagonal. As such, we can reduce matrix $\Sigma$ into $\Sigma_k$ which is an $k \times k$ matrix containing only the *k* singular values that we keep, and also reduce *S* and $U^T$, into $S_k$ and $U_k^T$, to have *k* columns and rows, respectively. Of course, all these matrix parts that we throw out would have been zeroed anyway by the zeros in $\Sigma$. Matrix *A* is now approximated by:

$$A_k = S_k \Sigma_k U_k^T \ . \tag{4}$$

Observe that since $S_k, \Sigma_k$ and $U_k^T$ are $m \times k$, $k \times k$, and $k \times n$ matrices, their product, $A_k$ is again an $m \times n$ matrix. Intuitively, the *k* remaining ingredients of the eigenvectors in *S* and *U* correspond to *k* “hidden concepts” where the terms and documents participate. The terms and documents have now a new representation in terms of these hidden concepts.

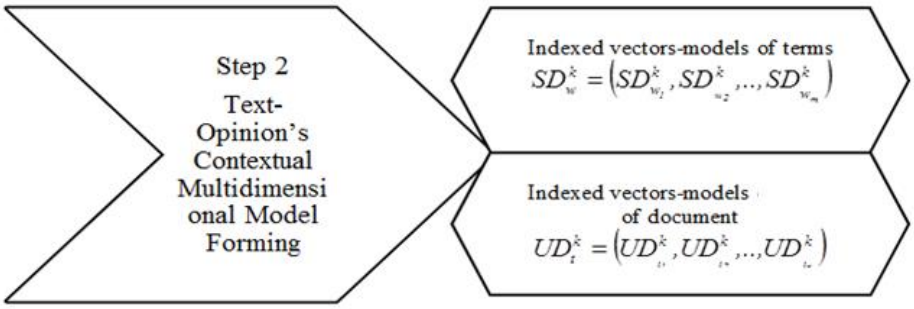
Namely,

– the terms are represented by the row vectors of the *m* × *k* matrix $S_k \Sigma_k$, which contain terms coordinates if the *k*-dimensional space $SD_w^k = \left(SD_{w_1}^k, SD_{w_2}^k, .., SD_{w_m}^k\right)$;

– the documents are represented by the column vectors of the *k* × *n* matrix $\Sigma_k U_k^T$ which contain documents coordinates if the *k*-dimensional space $UD_t^k = \left(UD_{t_1}^k, UD_{t_2}^k, .., UD_{t_n}^k\right)$.

Thus, the basic idea of *LSA* is that a matrix $A_k$ containing only the *k* first linearly independent components $A$ and represents the basic structure of the associative dependency of the original matrix, and at the same time does not contain noise.

As a **result of the second step of the** Dimensional Modeling of Client Text-Opinions Space: for text-opinions collection we have the **K-dimensional model** represented by the array of indexed vectors-models in the common *k*-dimension space (fig. 7):


Step 2
Text-Opinion's Contextual Multidimensional Model Forming
Indexed vectors-models of terms
$SD_w^k = \left(SD_{w_1}^k, SD_{w_2}^k, \dots, SD_{w_m}^k\right)$
Indexed vectors-models of document
$UD_t^k = \left(UD_{t_1}^k, UD_{t_2}^k, \dots, UD_{t_n}^k\right)$

.

Fig. 7. The results of the second step of the Dimensional Modeling of Client Text-Opinions Space

## 3.3. The Text-Opinion's Documents Semantic Clustering Model Forming

To compute "similar" documents to the given query we need to use the specific concept of the **semantic clustering algorithm of the customers text-opinions collection** $f(dist_t)$**, based on the** author's interpretation of the standard distance interpretation in conjunction with the LSA method results. This algorithm contains the following phases:

### Phase 1: The reference point on the scale measuring the closeness degree of the customer's text-opinions collection determination

As a *point of reference* in the scale of the customer's text-opinions collection closeness determination can serve to coordinate the *Central-frequency* (CF) term $CF^k = \left(SD^1, SD^2, \dots, SD^k\right)$ (group of terms) with the highest total weight [16]:

$$D_{w_i}^{\mathrm{CF}} = max\{\sum_{i=1}^{n} D_{w_i}\}. \tag{5}$$

The high weight of this term among the customer's text-opinions collection indicates the presence of at least one cluster, the center of which is this *Central-frequency* term (term group).

### Phase 2: The closeness degree of the documents identification

As an instrument of the documents' closeness degree identification should be used the *reference dimensional coordinate* $dist_{t_i}$ – the distance between the indexed vectors-models of the documents and coordinates of a point of reference in the scale of the customer's text-opinions collection closeness determination.

### Phase 3: The measures of similarity between pairs of documents recognition

The *measure of similarity* $K_{i+1,i}$ between pairs of documents is justified to consider the difference between the values of their relative spatial coordinates $dist_{t_i}$ :

$$K_{i+1,i} = dist_{i+1} - dist_{i}\,. \tag{6}$$

While documents shall be sorted in ascending order of values $dist_{t_i}$ .

Taking into account the heuristics introduced in consideration with determining the relative dimensional coordinates $dist_{t_i}$ , it is encouraged to use the following **author's concepts of standard distance metrics**:

Euclid distance:

$$dist_{t_i} = E(t_i, CF) = \sqrt{\sum_{i=1}^{k}\left(UD^{i}_{t_i} - SD^{i}\right)^{2}}\ , \tag{7}$$

where $UD^{k}_{t} = \left(UD^{k}_{t_1}, UD^{k}_{t_2}, .., UD^{k}_{t_n}\right)$ – indexed vector-model of the text-opinion document, $SD^{k}_{w} = \left(SD^{k}_{w_1}, SD^{k}_{w_2}, .., SD^{k}_{w_m}\right)$ – indexed vector-model of the terms.

Cosine similarity measure:

$$dist_{t_i} = cos(t_i, CF) = \frac{\sum_{i=1}^{k} UD^{i}_{t_i} \bullet SD^{i}}{\left\| \sqrt{\sum_{i=1}^{k}(UD^{i}_{t_i})^{2}} \right\| \bullet \left\| \sqrt{\sum_{i=1}^{k}(SD^{i})^{2}} \right\|} \tag{8}$$

Then the cosine of the angle between the documents vectors, which are sorted in ascending order of values $dist_{t_i}$ will be determined according to the relationship:

$$\begin{gathered} cos((t_{i+1}, CF) - (t_i, CF)) = \\ cos(t_{i+1}, CF) \cdot cos(t_i, CF) + \sqrt{1 - cos^{2}(t_{i+1}, CF)} \cdot \sqrt{1 - cos^{2}(t_i, CF)} \end{gathered} \tag{9}$$

By using the non-negative words' weights the cosine measure of similarity between the pairs of documents takes values in the range [0, 1], so for evaluation of the vectors difference the following formula are used:

$$K_{i+1,i} = 1 - cos((t_{i+1},CF) - (t_i,CF)) \, . \qquad (10)$$

## Phase 4: The Text-Opinion's Documents Semantic Clustering Model Forming

In order to form a Text-Opinion's Documents Semantic Clustering Model the following transformations presupposed:

1) Implementing the procedure of the partition of similarity measure $K_{i+1,i}$ into clusters using the *k*-means algorithms.
2) Contextual cluster's description forming using the modified function of *TF/IDF* transformation, which evaluates the term's preference for its inclusion in contextual cluster's description [2]. This measure takes into account the frequency of occurrence of key-words within a cluster compared to its occurrence in the documents of other clusters)

$$D^C_{w_i} = TFIDF^C_N(w,t) = \frac{k(w,L_C)}{S(L_C)} \bullet \frac{C_N}{C_N(w)} \bullet \frac{N}{N(C)} \bullet \frac{1}{1+log(1+\sigma^C_w)} \qquad (11)$$

where $k(w,L_C)$ – frequency of occurrence of term *w* in the text-opinion of the cluster *c*; $S(L_C)$ – term's number in the cluster *c*; $C_N$ – clusters number; $C_N(w)$ – number of the clusters, in which term *w* is occurred; $N$ – number of the text-opinion documents in the cluster *c*, in which term *w* is occurred; $N(C)$ – number of the text-opinion documents in the cluster *c*; $\sigma^C_w$ – the average deviation of the frequency of term *w* occurrence in the text-opinion documents of the cluster *c.*

As a **result of the third step** of the methodology of **semantic clustering of the customer's text-opinions collection** implementation we have the combination of the following elements (fig. 8):

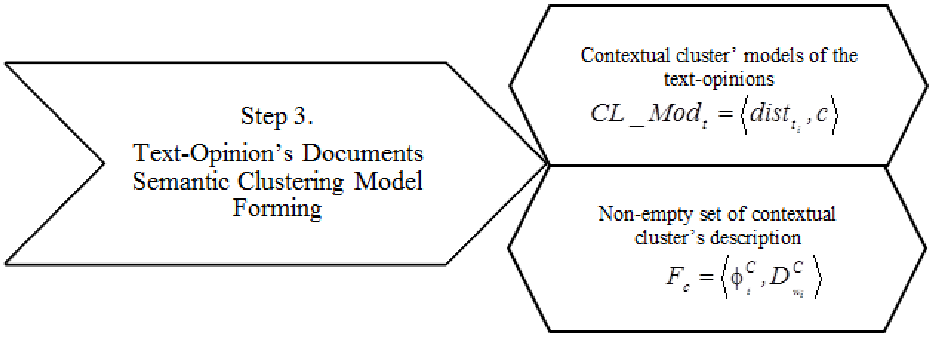

Step 3.
Text-Opinion's Documents Semantic Clustering Model Forming
Contextual cluster' models of the text-opinions
$CL\_Mod_t = \langle dist_{t_i}, c \rangle$
Non-empty set of contextual cluster's description
$F_c = \langle \phi^C_t, D^C_{w_i} \rangle$

Fig. 8. The results of the third step of the Dimensional Modeling of Client Text-Opinions Space

## 3.4. The Clients Opinion Semantic Communities Model Forming

The aim of the Clients Opinion Semantic Communities Model Forming is the clients groups (communities) revealing within each cluster.

The process of these groups revealing is based on the systematic analysis of:

- *contextual semantic similarity* of the customers, whose opinions are in the same cluster (the result of the steps 1-3 realization);
- *semantic structural similarity* of the customers, whose opinions are in the same contextual cluster – clients opinion semantic communities finding.

Development of the algorithm for determining the clients opinion semantic communities on the bases of the *semantic structural similarity* of customers opinions' presupposed existence of the following *hypothesis*:

*Correlation* $r_{ij}$ *between the one-dimensional vectors of the terms frequency identifications of the documents* $D^C_{t_i} = \left(D^C_{w_i}, D^C_{w_2}, ..., {}^{C}_{D_{w_{St}}}\right)$ *indirectly characterizes the semantic structural similarity of customer's style of thinking, and thus – preferences in the choosing and evaluation of products.*

As a mathematical apparatus evaluation of this phenomenon is proposed to use the instruments of the theory of *Social Network Analysis*, directed at revealing the features of objects on the basis of information about structure and strength of their interaction.

This algorithm assumes implementation of the following:

On the basis of $TFIDF^C_N$ frequency matrix the correlation matrix of dependencies between the analyzed documents inside the cluster $W^C = \{r_{ij}\}$ is formed. As the basis for the evaluation of the level of dependence between documents we use the measure of similarity between frequency characteristics of the common words, found in the documents.

This matrix is the basis for applying the modularity optimization algorithm of the SNA theory.

Modularity is a metric that was proposed by Newman and Girvan in reference [10, 19]. It quantifies the quality of a community assignment by measuring how much more dense the connections are within communities compared to what they would be in a particular type of random network. One of the mathematical definitions of modularity was proposed in the original paper on the Louvain method [1, 5, 19]:

$$Q = \frac{1}{2m} \sum_{i,j} \left[ P_{ij} - \frac{k_i k_j}{2m} \right] \delta\left(c_i, c_j\right) \tag{12}$$

Here, $P_{ij} = W^C$ is the weighted adjacency matrix, $k_i = \sum_j r_{ij}$ is the total link weight penetrating node *i*, and $m = \frac{1}{2}\sum_{i,j} r_{ij}$ is the total link weight in the network overall. The Kronecker delta $\delta(c_i, c_j)$ is 1 when nodes *i* and *j* are assigned to the same community and 0 otherwise. Consider one of the terms in the sum. Remember that $k_i$ is the total link weight penetrating node i, and note that $\frac{k_j}{2m}$ is the average fraction of this weight that would be assigned to node j, if node i assigned its link weight randomly to other nodes in proportion to their own link weights). Then, $P_{ij} - \frac{k_i k_j}{2m}$ measures how strongly nodes i and j are in the real network, compared to how strongly connected we would expect them to be in a random network of the type described above.

In our case, we can transform the classical definition of the Modularity metric as the following: *Molularity is number of edges, which are the correlation relationships between the documents inside the particular Cluster on the bases of the co-occurrence of the words, falling within documents with common context (clusters) minus the expected number in an equivalent documents network with edges placed at random.*

The problem of modularity optimization can be thought of as assigning communities such that the elements of the sum that contribute are as positive as possible. Larger modularity *Q* indicates better communities. *Q~=0.3-0.7* indicates good partitions.

As a **result of the fourth stage** of the methodology of **semantic clustering of the customer's text-opinions collection** implementation we have the intra-clusters clients opinion semantic communities, which contain the set of the clients $\phi_t^C$ (text-opinions authors) and communities' labels $Mod_{t_i}^C$ (fig 9).

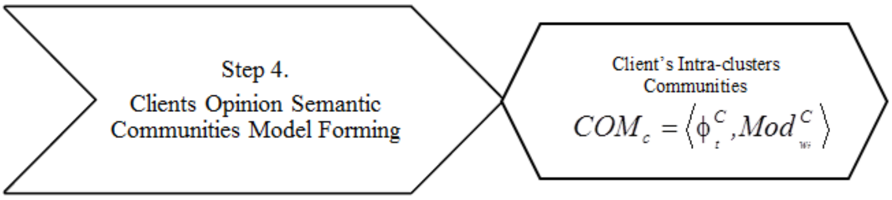

Step 4.
Clients Opinion Semantic Communities Model Forming
Client's Intra-clusters Communities
$COM_c = \langle \phi_t^C, Mod_{w_i}^C \rangle$

Fig. 9. The results of the fourth step of the Dimensional Modeling of Client Text-Opinions Space

## 4. THE OPERATORS

The operators work on data that has usually form of a free text containing opinions or beliefs about other database-entities (e.g. customer opinions about products) (fig.10).

```
<subsetting> := <subsetting>
      | 'where' <dimension> 'about' <subject> 'is' <object>
<aggregation> := <aggregation>
      | 'by' 'similar' <dimension>
<dimension> := <dimension> | 'keywords' 'of' <dimension>
```

Fig. 10. EBNF syntax for ADA NQL Extensions

As a sample for research and testing of the Text-mining Similarity Approximation Operators for Opinion Mining were used the database of the text-opinions collection of the Starbucks coffee shop customer's network (https://www.trustpilot.com/review/www.starbucks.com). In this stage of research we proposed the results of testing the Text-mining Similarity Approximation Operators using the special program for linguistic analysis, developed by the authors in Python.

This experiments helps us to study the possibilities of receiving the input data (as a result of the Dimensional Modeling of Client Text-Opinions Space) for processing it by complex Natural Language Processing.

## 4.1. "Keywords of" dimension

Dimension definition "*keywords*" is based on the results of the *Text-Opinion's Documents Semantic Clustering Model Forming*. Within ADA grammar access to keywords (the sets of the contextual cluster' models of the text-opinions $CL\_Mod_t = \langle dist_{t_i}, c \rangle$ and contextual cluster's description $F_c = \langle \phi_t^C, D_{w_i}^C \rangle$) is realized with dimension definition "keywords of C", where C is other dimension that contain opinions (Figure 10).

It was assumed that the maximum possible (experts given) number of clusters generated as a result of this method, is equal to 5 and includes the following contextual interpretation:

C={*C_1="Very good", C_2="Good", C_3="Satisfactorily", C_4="Bad", C_5="Does not correspond to the topic"*}.

The example of the experimental results of the queries "keywords by similar opinion" realization presented in the Table 1:

Table 1. The example of the experimental Contextual cluster's description results

| ***C_1*** | | ***C_2*** | | ***C_3*** | | ***C_4*** | |
|---|---|---|---|---|---|---|---|
| Keywords | $D_{w_i}^C$ | Keywords | $D_{w_i}^C$ | Keywords | $D_{w_i}^C$ | Keywords | $D_{w_i}^C$ |
| love | 0,22 | customer | 0,19 | not | 0,14 | service | 0,121 |
| time | 0,14 | very | 0,15 | nothing | 0,12 | wait | 0,1 |
| good | 0,12 | shop | 0,13 | make | 0,1 | say | 0,09 |
| always | 0,1 | well | 0,12 | because | 0,09 | order | 0,074 |
| me | 0,09 | like | 0,09 | unfortunately | 0,06 | visit | 0,067 |
| friend | 0,05 | all | 0,07 | went | 0,04 | something | 0,052 |

| | | | | | | |
|---|---|---|---|---|---|---|
| | fan | 0,02 | want | 0,03 | get | 0,043 |
| | | | buy | 0,02 | from | 0,031 |
| | | | back | 0,017 | week | 0,027 |
| | | | order | 0,014 | thing | 0,02 |

The obvious characteristics of the received Contextual cluster's description are the following:

- significant increase in the terms (key-words) number of in clusters with negative (Satisfactorily and Bad*)* evaluations of the product;
- reducing of the key-words weight in clusters with negative evaluations, indicating by less uniform distribution of the key-words in such opinions and, including the fact that the negative opinions are often overwhelmed by not related to the topic of the survey, emotions, gives examples of incidents from their own life and only casually refer to the main topic.

## 4.2. "About is" subsetting operator

Subsetting operator "about C is q" is also based on the results of the *Text-Opinion's Documents Semantic Clustering Model Forming.*

Having the sets of the contextual cluster' models of the text-opinions $CL\_Mod_t = \left\langle dist_{t_i}, c \right\rangle$ and contextual cluster's description $F_c = \left\langle \phi_t^C, D_{w_i}^C \right\rangle$ we can compute the representation of a "query" by computation of the centroid (mean vector) of the vectors for its terms. The operator has the semantics defined in Figure 5 and is interpreted as : <dimension> – dimension with opinion, "query" = (<subject>,<object>).

The example of the experimental data for processing the queries of the following format: "*Count shops by country where opinion about coffee is good on a map*" presented in the Table 2 and fig.11. On the bases of this data and additional information about shops location system can give the answer to such questions.

Table 2. The example of the experimental results of the Text-Opinion's Documents Semantic Clustering Model Forming

| Number of text-opinions documents in clusters (%) | | | | |
|---|---|---|---|---|
| C_1 | C_2 | C_3 | C_4 | C_5 |
| 20,69% | 17,24% | 24,14% | 31,03% | 6,90% |

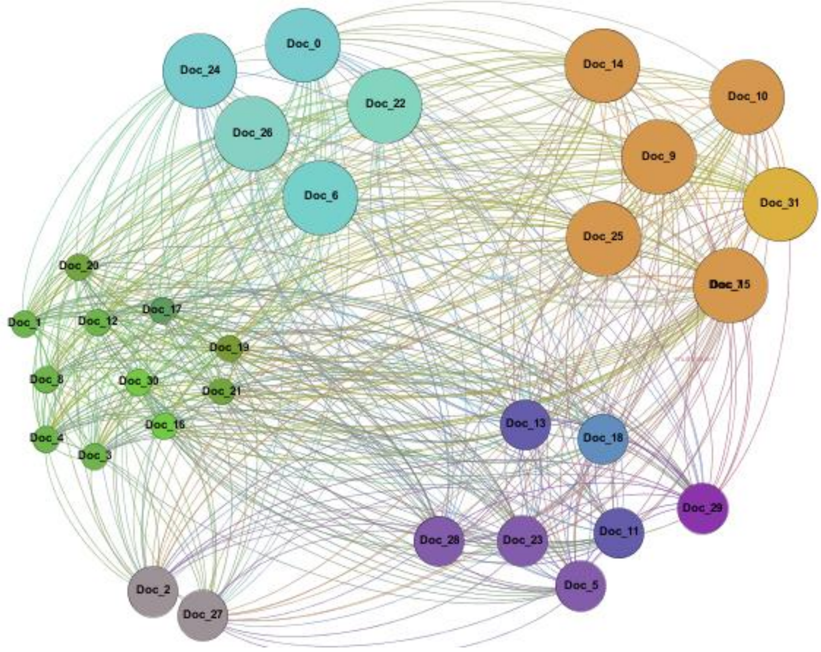


Fig. 11. The example of the experimental visualization of the of Text-Opinion's Documents Semantic Clustering Model Forming results

### 4.3. "By similar" aggregation operator

Aggregation operator "by similar C" is based on the results of *The Clients Opinion Semantic Communities Model Forming* stage.

The operator has the semantics defined in Figure 5 and <dimension> is a dimension with opinion. Communities that are results of the applied operator are grouped together and allowed to be aggregated.

The examples of the experimental data for processing and visualization of the queries of the following format: "*count clients by similar opinion*" presented in the Table 3 and fig.12. On the bases of this data and additional information about shops location system can give the answer to such questions.

Table 3. The experimental example of the Clients Opinion Semantic Communities Model Forming results for cluster with "Bad opinions"

| Number of clients in community (%) | Keywords $\phi_t^C$ | Cluster's label $Mod_{w_i}^C$ |
|---|---|---|
| 45% | service, wait, say, order | Unsatisfied with the service |
| 55% | visit, something, get, from, week, thing | Unsatisfied by other things |

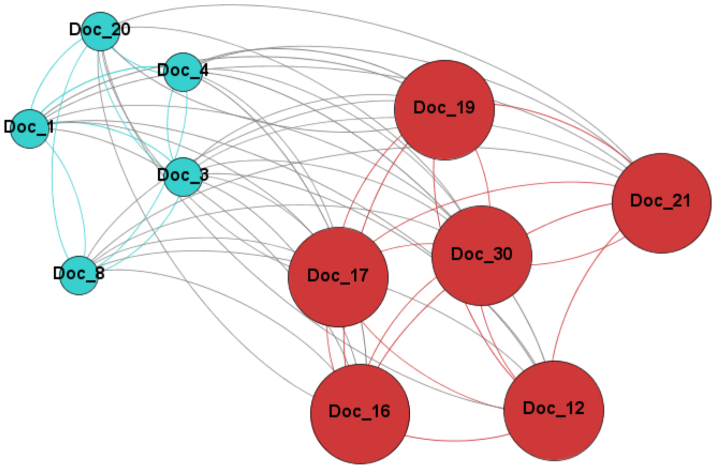


Fig. 12. The experimental example of the Clients Opinion Semantic Communities Model Forming stage results visualization (communities of the cluster with "Bad" opinions)

## 5. CONCLUSIONS

In this paper the concept of the Text-mining Similarity Approximation Operators for Opinion Mining as extensions to NLIDB, called Data Anything, are defined. For the implementation of the proposed concept in BI tools four preprocessing steps of the Dimensional Modeling of Client Text-Opinions Space are suggested and described. As an extension of the Natural Language Interface Database (NLIDB) the idea of the new operators was developed:

- *"keywords of" dimension* and *subsetting operator "about C is q"* – based on contextual cluster' models of the text-opinions and contextual cluster's description;
- *aggregation operator "by similar C"* – based on the set of the clients and communities' labels as a results of the SNA Modularity Optimization.

The results of experimental research of proposed operators as a background of the Text-mining Similarity Approximation Operators for Opinion Mining are also presented. As a sample for those experiments text-opinions collection of the Starbucks coffee shop customer's network is used.

In future work, we look forward to proposing a more extended application of the new operator's in the BI system – directly at the ADA-tools experimental level. Additionally, we hope to integrate the Latent Semantic Social Network Analysis with and recommendation module level.

## BIBLIOGRAPHY

[1] Ahuja R.K., Magnanti T.L., and Orlin J.B., "Network Flows: Theory, Algorithms, and Applications". Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1993.

[2] Andreev A., Berezkin D., Morozov V., Simakov K.. “The method of clustering texts collections and clusters annotating”. Digital Libraries: Advanced Methods and Technologies, Digital Collections: Proceedings 10th Scientific Conference (RCDL'2008). – Dubna, 2008. pp. 220-229.

[3] Bollobas B., “Modern Graph Theory, ser. Graduate Texts in Mathematics”. Springer New York, 1998. [Online]. Available: https://books.google.pl/books?id=SbZKSZ-1qrwC

[4] Bradford R.B., “An empirical study of required dimensionality for large-scale latent semantic indexing applications,” in Proceedings of the 17th ACM Conference on Information and Knowledge Management, ser. CIKM ’08. New York, NY, USA: ACM, 2008, pp. 153–162. [Online]. Available: http://doi.acm.org/10.1145/1458082.1458105

[5] Clauset A., Newman M.E.J., and Moore C., “Finding community structure in very large networks,” Physical Review E, pp. 1– 6, 2004. [Online]. Available: www.ece.unm.edu/ifis/papers/community-moore.pdf

[6] Dobrowolski D., Kaplanski P., Marciniak A., and Lojewski Z., “Semantic OLAP with FluentEditor and Ontorion Semantic Excel Toolchain,” IARIA, vol. SEMAPRO 2015: The Ninth International Conference on Advances in Semantic Processing, 2015. [Online]. Available: https://www.thinkmind.org/index.php?view= article&articleid=semapro_2015_3_30_30051

[7] Dumis S, Fumas G, Landauer T et al. “Using Latent Semantic Analysis to Improve Access to Textual Information”. Proceedings of Computer Human Interaction, 1988.217-285

[8] Hofmann T., “Probabilistic latent semantic indexing,” in Proceedings of the 22Nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, ser. SIGIR ’99. New York, NY, USA: ACM, 1999, pp. 50–57. [Online]. Available: http://doi.acm.org/10.1145/312624.312649

[9] Jurgens D. and Stevens K., “The s-space package: An open source package for word space models,” in Proceedings of the ACL 2010 System Demonstrations. Uppsala, Sweden: Association for Computational Linguistics, July 2010, pp. 30–35. [Online]. Available: http://www.aclweb.org/anthology/P10-4006

[10] Newman M. E. J. and Girvan M., “Finding and evaluating community structure in networks,” Physical Review, vol. E 69, no. 026113, 2004.

[11] Nolan C., “Manipulate and query olap data using adomd and multidi-mensional expressions.” Microsoft Systems Journal, no. 63, pp. 51–59, 1999.

[12] Øehùøek R. and Sojka P., “Software framework for topic modeling with large corpora,” in Proceedings of LREC 2010 workshop New Challenges for NLP Frameworks. Valletta, Malta: University of Malta, 2010, pp. 46–50. [Online]. Available: http://www.fi.muni.cz/usr/sojka/presentations/lrec2010-poster-rehurek-sojka.pdf

[13] Pedersen T., “Duluth : Word sense induction applied to web page clustering,” in Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation

(SemEval 2013). Atlanta, Georgia, USA: Association for Computational Linguistics, June 2013, pp. 202– 206. [Online]. Available: http://www.aclweb.org/anthology/S13-2036

[14] Ranta A., Controlled Natural Language: 4th International Workshop, CNL 2014, Galway, Ireland, August 20-22, 2014. Proceedings. Cham: Springer International Publishing, 2014, ch. Embedded Controlled Languages, pp. 1–7. [Online]. Available: http://dx.doi.org/10.1007/ 978-3-319-10223-8_1

[15] Rehurek R., "Subspace tracking for latent semantic analysis," in Advances in Information Retrieval - 33rd European Conference on IR Research, ECIR 2011, Dublin, Ireland, April 18-21, 2011. Proceedings, 2011, pp. 289–300. [Online]. Available: http://dx.doi.org/10.1007/ 978-3-642-20161-5_29

[16] Rizun N., Kapłanski P., Taranenko Y. "Development and Research of the Text Messages Semantic Clustering Methodology", The Third European Network Intelligence Conference (ENIC 2016), Proceedings, 2016.

[17] Salton G, Wong A, Yang CS. "A Vector Space Model for Automatic Indexing". Communications of the ACM, 1995,18(11): pp 613-620.

[18] Seganti A., Kaplanski P., Campo J.D.N, Cieslinski K., J. Kozi-olkiewicz, and P. Zarzycki, "Asking data in a controlled way with Ask Data Anything NQL," vol. CNL2016 Conference. Springer, 2016.

[19] West D., Introduction to Graph Theory, ser. Featured Titles for Graph Theory Series. Prentice Hall, 2001. [Online]. Available: https://books.google.pl/books?id=TuvuAAAAMAAJ

[20] Xuren Wang, Qiuhui Zheng. "Text Emotion Classification Research Based on Improved Latent Semantic Analysis Algorithm". Proceedings of the 2nd International Conference on Computer Science and Electronics Engineering (ICCSEE 2013)