EasyChair Preprint

№ 3157

Search Query Data Analysis: Challenges and Opportunities

Olena Pavlenko and Iryna Tymofieieva

April 12, 2020

Search Query Data Analysis: Challenges and Opportunities

Olena Pavlenko [0000-0002-3747-4651], Iryna Tymofieieva [0000-0002-5935-9291]

Faculty of Foreign Languages
Mariupol State University, Mariupol, Ukraine

h.pavlenko@mdu.in.ua, tib_kalmius@ukr.net

Abstract. In the light of tremendous growth of the amount of information available in the Internet over the last decade, effective use of query data technologies have become the basis of competition. The article addresses the issues of how to gain a deeper understanding of the technology trends to effectively deal with large volumes of data as well as use algorithms to provide the accuracy of results. It then considers the prospects and challenges regarding the user's ability to summarize the essential properties of data retrieval, and puts forward appropriate techniques to achieving these objectives focusing on the natural language processing in information retrieval. Following that, the paper presents a comprehensive approach for search queries taking into account the wider content of search based on the collection, transmission, storage analysis, storage and display of information.

Keywords: Internet resource; cluster; semantic search.

1. Introduction

Search engines are currently recognized not only as a complex of information storage programs equipped with an advanced reference search engine, but also as dominant statistical and analytical systems.

The underlying assumption in this case is the experience of the popular Google Trends service, respectively owned by Google. The advantage of this service is not only in identifying the most stable popular queries in general, which can be obtained with the help of most traditional Internet search engines, but also in obtaining statistics for a short period of time - the last 24 hours or a specified period of time in the past. In the last decade, there has been a flurry of work on evaluating information extraction systems based on a performance problem, but the methods above do not prove to be sufficient for systems. In addition, for ease of use, we must consider such aspects as scalability, availability, and interoperability.

2. Analysis Of Recent Researches And Publication

Using computers to understand and operate natural language text to perform useful things and desired assignments has become quite prominent due to the enormous increase of the World Wide Web and digital libraries. Following this, the article [1], describing methods and basic issues of normalizing the text, is of great theoretical value for our research. Thus, the authors assume that one major challenge in text normalization research has been the lack of annotated data for training and evaluating methods. As a result, most Twitter text normalization methods have been unsupervised or semi-supervised (Cook and Stevenson, 2009; Han et al., 2012; Yang and Eisenstein, 2013), and evaluated over small-scale hand annotated datasets. This has hampered analysis of the strengths and weaknesses of individual methods, and was our motivation in organizing the lexical normalization shared task.

For example, GoogleTrends Service involves conducting a “vertical search” of 24 thematic categories and 10-30 subheadings. Unfortunately, they cover the most common consumers’ needs, such as "vehicles" or “Business and Industry”, but fail to set a search area for narrow scientific, social, and especially library science issues.

From the “Science” column, the service distinguishes only 11 sub-sections related only to natural science disciplines, scientific equipment and scientific organizations.

We can search multivalent words used both in library science as well as in other disciplines by referring to the tags that pop up when typing a word in the search box. Thus, you can search the word “library” just clicking on the tag:

- “search query”;
- “topic”;
- “programming”;
- “public libraries’ system”.

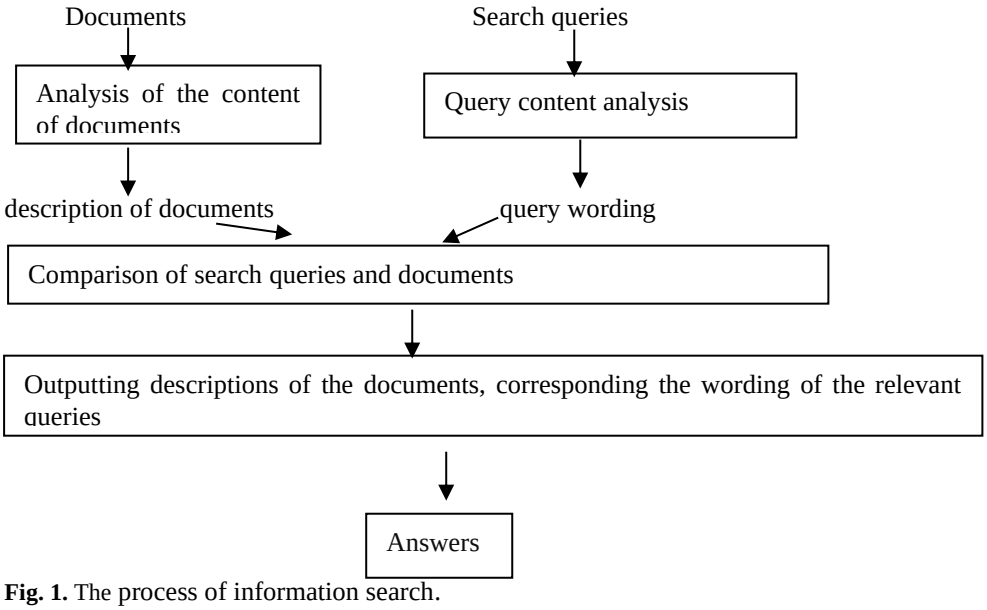

Fig. 1. The process of information search.

Pablo Barcelo Baeza, a researcher from the University of Chile [2] studies the problem of querying graph databases, and, in particular, the expressiveness and complexity of evaluation for several general-purpose query languages, such as the regular path queries and its extensions with conjunctions and inverses. He distinguishes between two semantics for these languages. The first one, based on simple paths, easily leads to intractability, while the second one, based on arbitrary paths, allows tractable evaluation for an expressive family of languages.

A list for selecting different tags is set automatically, depending on the number of topics and categories affiliated with a keyword in a search engine that can affect the user. However, to eliminate the artificial limitation of the search we recommend searching for narrow specialized keywords for all thematic headings [3].

In their works [4], the authors stated traditional IE is not, however, completely suitable for metadata creation, because semantic tags need to be mapped to instances of concepts, attributes or relations, and this is not always a straightforward process.

There are no separate scholarly monograph to provide a detailed description of the GoogleTrends service capabilities for solving source search problems. We focus on the research of Ben Showers "Library Analytics and Metrics: Using Data to Drive Decisions and Services" [5]. In a chapter "External toolkit for researching the behavior of visitors of public information and cultural institution", GoogleTrends is described in detail with examples of the correlation between the popularity of the concepts of the British Museum and the Terracotta Army, in order to make a conclusion about the popularity of a particular exposition among all other museum exhibitions.

While researching the problem we based on the article [6], which describes the construction of a new algorithm for embedding words. Most word embedding algorithms only use the word pairs that occur in the corpus (i.e. positive examples) and maximize the similarity of those word vectors based on how frequent they co-occur. This can result in a concentration effect: word clusters from totally different topics can be placed somewhat close to each other. There are lots of possible word pairs that never co-occur in the corpus or they co-occur insignificantly, which we call negative examples. We argue that minimizing the similarity of negative examples is also crucial for the quality of the final embedding and results in a better distribution of words in the latent space. Our first major contribution is to design an optimization framework that exploits the full capacity of negative examples in order to push unrelated words away from each other, which leads to a better use of the latent space and improves the distribution of words. Our second major contribution is that we incorporate a kernelized weighting scheme for the negative examples where their influence in the optimization is proportionate to their kernel similarity with the word. We show that our kernel similarity measure is a more powerful way of calculating similarities in high-dimensional embeddings where $d > 50$ and it enables the algorithm to differentiate between the closer and further points and employ them accordingly.

We also steer our priorities to support the researchers' ideas [7; 8], that the ability to plug and play is an essential step in standardization. Having the ability to plug outputs and inputs in a query language incentivizes its adoption (modularity, interoperability); simplify abstractions, users do not have to think about multiple data models during the query process; and increases its productivity, by facilitating reuse and decomposition

of queries. These researches provide the evidence to conclude, that basic graph query consists of a single Construct clause followed by one Match clause. The G-CORE query language has been carefully designed to ensure that G-CORE queries can be evaluated e efficiently in data complexity. Formally, this means that for each fixed G-CORE query q, the result [g] G of evaluating q over an input PPG G can be computed in polynomial time.

3. The Main Tasks Of The Study And Their Significance

Current query languages do not provide full composability because they output tables of values, nodes or edges. In view of the conducted analysis, we assume that the set of used approaches cannot claim to be versatile and search optimization remains an actual task. One of the main goals of this paper is to provide a formal definition of the syntax and semantics of a graph query language including the features shown in the previous sections.

User search queries become the most important source of information for Internet resources owners. Since the findings of the analysis of search queries can potentially improve the quality of the search, as they help to better understand the interests of users. However, taking into account the knowledge gained, search engines will show the most relevant documents to the user. One of the main problems in analyzing search queries is the ambiguity of the words used in them. One classic example of this ambiguity is the “jaguar” query. In this case, it is unclear what information were the users looking for: about cars or animals. If we have knowledge about the interests of the user who entered an ambiguous request, we can easily determine what kind of information he wanted to know.

Much attention is also paid to the methods that allow you to convert an unstructured user query with "keywords queries" into a structured one. The main reason for the popularity of such methods is that most of the Internet data is initially contained in structured databases. Moreover, the knowledge of the request structure greatly facilitates the search for relevant answers. To train a query analysis model that receives a structure from a query, we find it necessary to compose a training set in which each query is described by feature vectors or simply features – sets of numerical parameters that reflect the properties of the characteristics of the query.

Feature vectors take values in feature space. By setting a metric in such a space, you can compare queries with each other, calculating the distance between the vectors corresponding to them. Methods for creating a training set and constructing feature vectors are the core of any query analysis system. The quality of the search query analysis system mainly depends on the choice of the training set and features, as well as metrics for their comparison. The traditional approach for creating a query analysis system is known as supervised learning, but this method seems rather laborious and expensive, since it requires a manual training set.

Keyword grouping is the division of keywords into homogeneous groups according to certain criteria aimed to increase relevance between keywords and advertisements. As a result, the quality of advertising campaigns improves and the cost of an ad click decreases.

We also based on the research of Julia Kiseleva, in particular, on her work "Methods of grouping and structuring search queries and their implementation" where the author asserts that when preparing advertising campaigns, grouping keywords takes a lot of time. Some experts use the "1 keyword - 1 ad group" approach, which eliminates the need for grouping and saves time. Nevertheless, it is not always convenient in using, especially with a large number of low-frequency keywords. If the clustering of keywords (the division into semantic groups) was performed, then the structure of the advertising campaign becomes more reasonable and easier to analyze. Keyword grouping consists of the following steps: loading a list of keywords into the program; breakdown of all keywords into unique words; marking the words on the basis you want to create a group; uploading results to a csv file. A unique word can be marked as:

The main word – if on its basis you need to create a group in any case;

Additional – if on the basis of this word it is desirable to create a group;

Filtering – if it is needed to delete all keywords that contain the selected word;

Minusing – if you want to delete all keywords that contain the selected word, and save the word in a separate list for later use as minus words in an advertising campaign.

Grouping phrases (clustering) is the combination of key phrases into one ad group. Its main task is to distribute many objects into groups called clusters. Inside each group, there should be "similar" objects, and objects of different groups should be as different as possible. The grouping gives the PPC specialist: announcements that are more relevant (for the phrases united by one group, you can show a specific advertisement tailored to these phrases - with your own message and / or landing page); A / B testing of ads (within the same group, you can use more than one ad at a time. They will be shown alternately, depending on the rotation settings, and the user will identify a more conversion / clickable); Combining statistics of key phrases that are supposedly similar in behavior (when there is little data at the level of key phrases, moving to a higher level – to the group level allows you to accept more reliable statistical results); control and ease of management (it is more convenient to operate with larger entities – groups of phrases: enable / disable / analyze. Most often, key phrases are grouped by the following parameters: morphological proximity – cognates are sent to one group; semantic / semantic proximity – synonyms are sent to one group; by user's intention – phrases are sent to the same group that mark the same user's intention [11].

4. Major Research Results

Grouping is necessary when creating new Internet sites, as well as when reorganizing the structure and optimizing existing websites. It is also used when creating and conducting contextual advertising campaigns. The objectives of grouping are: cleaning the semantic core of inappropriate keywords; grouping keywords to create a site structure or contextual ads; a better understanding of user intentions for ambiguous search queries; the selection of regional, commercial and other specific types of search queries; combining in one group search queries that differ greatly in spelling, but are identical in content or intention of the user; assessment of the need to separate similar search queries into different groups for separate promotion due to high competition; other tasks, that require

combination of keywords in groups, implying the same intention of the user searching the search engine for keywords above [12].

The difference between the described algorithm and the currently available search query clustering algorithms is the use of TOP-30 search results with assignment of unequal weight to each of the results, decreasing from 1 to 30 places in terms of hyperbolic function (fig.2).

In Figure 2, the position of the search result is plotted along the P axis, and the corresponding weight on the W axis. The weight of the first place is 3.0 units, 30th - 0.25 units. The calculations use floating-point numbers. The choice of this curve is due to the device of the search results. The first result is the most relevant (theoretically), and then the relevance of the results is reduced to zero (accidental hit in the TOP when similar words coincide or other factors). Accordingly, the coincidence in the search results for two different requests of the 1st issuing places means much more than the coincidence in the 20-30th places [12].

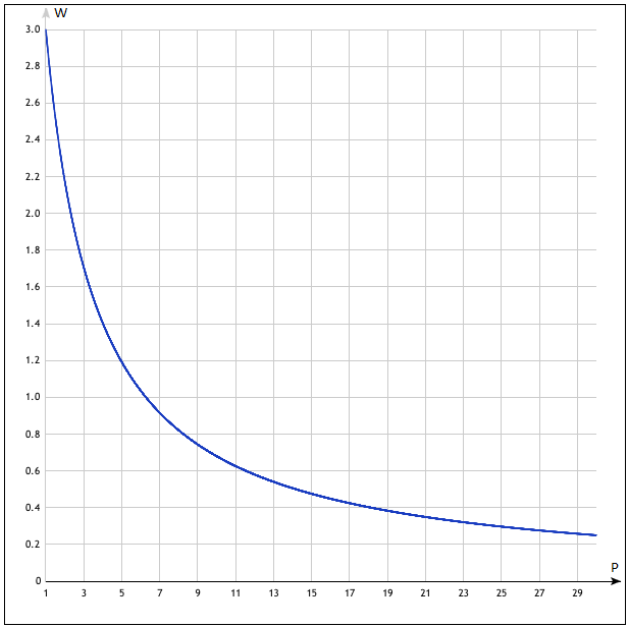

Fig. 2. Hyperbolic function.

To identify the exact coefficients by which the working hyperbole is built, we studied a lot of available sources and studies related to search results (its formation, CTR of different positions, etc.), as well as many automatic and manual clustering tests. The final formula for the hyperbola used in this algorithm is as follows (limits provided and there's still no guarantee that the matched odds are perfect):

$$W = 87/(11p + 18),\ p \in [1;\ 30]. \quad (1)$$

While considering a comparison of fingerprints of search results we use “Imprint of the issue” as a conditional term describing the algorithm. Issuing imprint (keyword) – is a list (table) of 30 first URL-addresses, that were taken as a result of parcing for a given search query. Each of these URLs in the print of the issue is assigned a weight corresponding to its position in the search results for this request. Weight is calculated according to the formula described above. The number of output fingerprints is equal to the number of requests in the semantic core. To compare two fingerprints of the issue, you need to find all the matching URLs in them. For each matching URL, there are values: p_1 and p_2 are the positions of this URL in the first and second print of the issue, respectively, and w_1 and w_2 are the corresponding weights. The values w_1 and w_2 of the matched URLs are multiplied and the square root is extracted from the resulting value. All results are summarized and compared with a threshold value. The final formula for comparing two prints of the issue is as follows:

$$\delta = \sum_{i=1}^{k} \sqrt{w_1^i w_2^i} = 87 \sum_{i=1}^{k} \sqrt{\frac{1}{(11p_1^i + 18)(11p_2^i + 18)}} \quad . (2)$$

Here k is the number of matching URL addresses (if k = 0, then $\delta = 0$). Important: “i” is a superscript, not an exponentiation [12].

The coincidence of two URLs depending on their position can be represented as a function

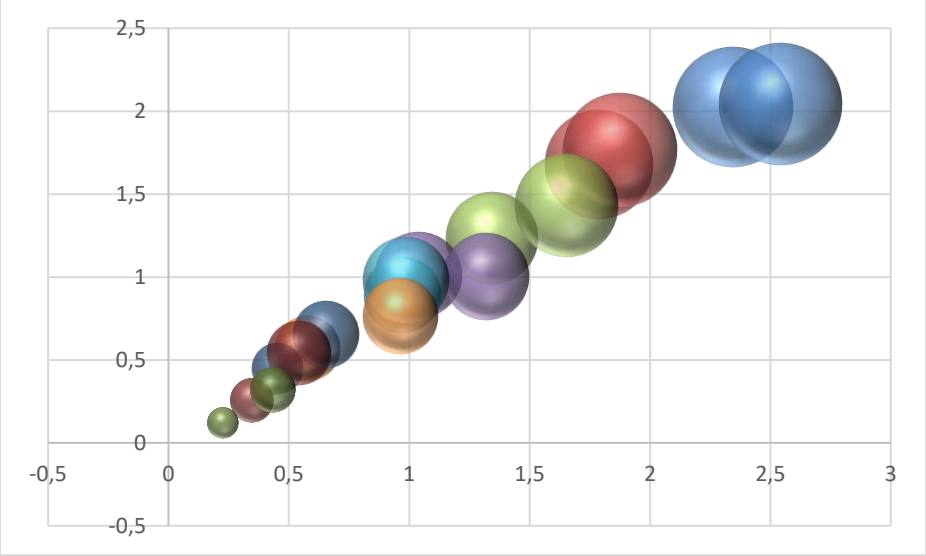

Fig. 3. f graphics.

The points in the graph correspond to the values obtained from fingerprint searching of the URL, with algorithm involving floating point numbers (fig.4).

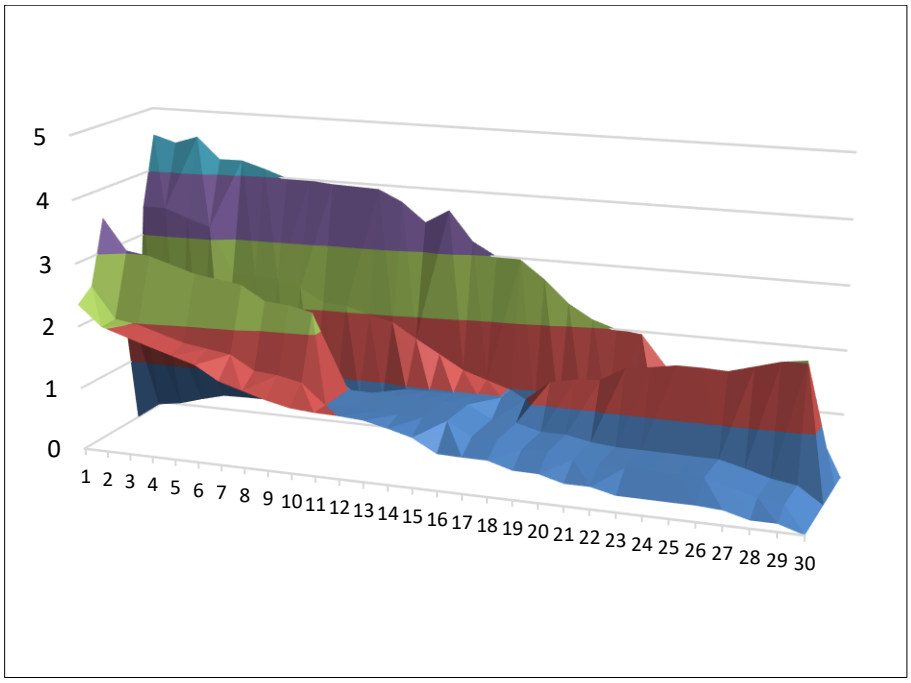

Fig. 4. f graphics.

As a result, you will need to find only the sum of the values from the prepared table:

$$\delta = \sum_{i=1}^{k} table\left[p_1^i\right]\left[p_2^i\right]. \qquad (3)$$

In particular, Rubenstein H., Goodenough J. [15] claim, that vector spaces from linear algebra are used as a way of representing the model. Information on the distribution of linguistic units is presented in the form of multi-bit vectors that form a verbal vector space. Vectors correspond to linguistic units (words or phrases), and dimensions correspond to contexts. The coordinates of the vectors are numbers showing how many times a given word or phrase has occurred in a given context.

The model of a verbal vector space that describes the distributive characteristics of the words he and she, in which the neighboring word is the context, is based on Sharnin's methodology

$$A_{m,n} = \begin{matrix} she \\ W_2 \\ he \\ \vdots \\ W_m \end{matrix}\begin{bmatrix} 0 & 1 & 0 & & 1 \\ 1 & 0 & 2 & \cdots & 0 \\ 0 & 2 & 0 & & 3 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 1 & \cdots & 0 \end{bmatrix}. \qquad (4)$$

The size of the context window is determined by the objectives of the study: the establishment of syntagmatic connections −1-2 words; the establishment of paradigmatic connections – 5-10 words; the establishment of thematic links – 50 words or more. The semantic proximity between linguistic units is calculated as the distance between vectors. In studies on distributive semantics, the cosine measure, which is calculated by the formula, is most often used [13]:

$$\frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \times \sqrt{\sum_{i=1}^{n} (B_i)^2}}. \qquad (5)$$

After conducting such an analysis, it becomes possible to identify the words that are closest in meaning to the word being studied.

We use terms xi as variables that will match any term in the поиске. In this case, the results will be as follows:

Table 1. Search results.

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
Madrid	Kiev	Paris	Helsinki	05.02.2019	15.02.2019		co		n_1
Paris	Helsinki	Madrid	Kiev	15.02.2019	05.02.2019	Po	nte	capit	n_2
Madrid	Kiev	Paris	Helsinki	05.02.2019	15.02.2019	st	nt lan	al eng	n_3
Paris	Helsinki	Madrid	Kiev	15.02.2019	05.02.2019		g	ua	n_4

5. Conclusion

In this paper, we investigated the issue of clustering (words grouping) analysis with clarification of temporary expressions and semantic links. We suggested comparing the search query based on semantic proximity identified as the distance between vectors. One of the fundamental characteristic features of our method is that, unlike SGNS, which uses each entry in the matrix to update the search word, our algorithm updates the word vector by means of semantic proximity based on its entire row in the matrix. The most time-consuming part of our algorithm comes to be the calculation of lexical group divergence that requires a set of calculation based on the similarity of words. In search for the solution to the word similarity problem, we suggest a number of data sets containing pairs of words with a corresponding similarity index specified by a user. Adaptive approach to use information retrieval prevents the effect of concentration and advances the distribution of words in the final display of search results.

The stream version of the word search algorithm lets words with affixes enter the procedure, thus making a vertex-oriented programming model one of the most visible and widespread word search algorithms. Further research in the field of analytics and information retrieval open up striking new possibilities applied in its potential format. We are also investigating the generation of networks of k-nearest words-“neighours” by their semantic similarity as well as considering issues to introduce a

search library extension that will provide infrastructure for presenting the algorithms specified by a user and guarantee their efficient implementation.

References

1. Baldwin, T., de Marneffe, M. C., Han, B., Kim, Y. B., Ritter, A., Xu, W.: Shared tasks of the 2015 workshop on noisy user-generated text: Twitter lexical normalization and named entity recognition. In: Proceedings of the Workshop on Noisy User-generated Text, pp. 126-135 (2015).
2. Barceló Baeza, P.: Querying graph databases. In: Proc. of the 32nd SYMPOSIUM ON PRINCIPLES OF DATABASE SYSTEMS (PODS '13) 2013, ACM, pp. 175–188 (2013).
3. Sokolov, S.: Applying Google Trends web-analytic tools in socio-humanitarian and library studies. Bibliosphere, no. 4, pp. 3–9 (2018).
4. Maynard, D.: Benchmarking ontology-based annotation tools for the Semantic Web, UK e-Science Programme All Hands Meeting (AHM2005) Workshop "Text Mining, e-Research and Grid-enabled Language Technology", Nottingham, UK, https://gate.ac.uk/sale/ahm05/ahm.pdf last accessed 2020/04/12.
5. Showers, Ben (ed.).: Library Analytics and metrics: using data to drive decisions and services. Facet Publishing. London UK, 176p. (2015).
6. Behrouz, H. S., Matwin S.: Fast PMI-based word embedding with efficient use of unobserved patterns. In: CONFERENCE 2019, The Thirty-Third AAAI Conference on Artificial Intelligence (AAAI-19), vol 33, no. 1, pp. 7031-7038 (2019).
7. Angles, R., Arenas, M., Barceló Baeza, P., Boncz, P., Fletcher, G., Gutierrez, C., Lindaaker, T., Paradies, M., Plantikow, S., Sequeda, J., Oskar Van Rest, Voigt.H.: G-CORE A Core for Future Graph Query Languages. CONFERENCE 2018, In: International Proceedings of the 2018 International Conference on Management of Data, pp. 1421-1432 (2018).
8. Angles, R., Arenas, M., Barceló Baeza, P., Hogan, A., Reutter, J., Vrgoč, D.: Foundations of modern query languages for graph databases. In: ACM Computing Surveys (CSUR). 2017/9/26, vol.50, no.5, pp. 1-40. New York, NY, United States (2017).
9. Fomichov, V., Kirillov, A.: Semantic Transformation of Search Requests for Improving the Results of Web Search. In: Pre-Conference Proceedings of the Focus Symposium on Intelligent Information Management Systems (August 2, 2011, Focus Symposia Chair: Jens Pohl) in conjunction with InterSymp-2011, 23rd International Conference on Systems Research, Informatics and Cybernetics, August 1 - 5, 2011, Germany). San Luis Obispo, CA, USA: Collaborative Agent Design Research Center, California Polytechnic State University (Cal Poly), pp. 37-43 (2011).
10. Lytvyn, V., Vysotska, V., Burov, Y., Bobyk, I., Ohirko, O.: The Linguometric Approach for Co-authoring Author's Style Definition. In: 2018 IEEE 4th International Symposium on Wireless Systems within the International Conferences on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS-SWS), vol. 6, no 2 (102), pp. 29-34 (2018).
11. Kiseleva, J.: Methods of grouping and structuring search queries and their implementation: the dissertation ... candidate of physical and mathematical sciences: 05.13.11, pp. 99. St. Petersburg (2011).
12. Dvurechensky, P.: Algorithm for hyperbolic query clustering, https://topsite-program.ru/blog/hyperbole, last accessed 2020/04/12.
13. Sharnin, M., Somin, N., Kuznetsov, I., Morozova, Yu., Galina, I., Kozerenko, E.: Statistical mechanisms of forming associative portraits of subject areas based on a large amount of

natural language texts for the system of knowledge extraction. Informatics and s applications: journal, vol. 7, no. 2, pp. 92-99 (2013).
14. Sketch Engine corpus manager. Lexical Computing, https://www.sketchengine.eu/, last accessed 2020/04/12.
15. Rubenstein H., Goodenough J.: Contextual correlates of synonymy. Communications of the ACM . Journal vol. 8, no. 10, pp. 627-633 (1965).