Extended Functionality of Mathematical Formulae Search Service

Alexander Gusenkov, Polina Gusenkova,
Yana Palacheva
IVMiIT, Kazan Federal University
Kazan, Russia
e-mails: alexandr.gusenkov@kpfu.ru
{polinagpa, palachevayana}@gmail.com

Olga Zhibrik
Gradient Technology Ltd
Kazan, Russia
e-mail: olgazhibrik@gmail.com

***Abstract*—This paper focuses on the Mathematical Formulae Search Service of the Lobachevskii-DML (Digital Mathematical Library) project. The service is based on the original method of mathematical document markup that allows establishing relations among terms, variables, and formulae. This method was tested in two different search services with different preprocessing approaches. In Lobachevskii-DML, the instances of the mathematical entities are elicited as ontology concepts. The search service enables the user to seek formulae by textual definitions of their variables by generating ontology queries in SPARQL (SPARQL stands for SPARQL Protocol and RDF Query Language, RDF is the Resource Description Framework). The paper provides an overview of the search service and discusses the dynamic generation of the queries in response to new functionality features, including seeking formulae by more than one ontology concept.**

Keywords-semantic search; mathematical knowledge; ontology; formulae markup.

I. INTRODUCTION

The well-known phenomenon of the rapid increase in the amount of published information in many scientific fields has led to a growth of interest towards the subject of information structuring. The ongoing global digitalization of all the existing hard copy sources makes the research in this field even more important. Usually, scientific documents have a specific structure which is defined by the field. In case of mathematical documents, the text contains formulae, symbolic notations, and terms for the entities of the field. In terms of context, mathematical texts can be divided into theorems, axioms, proofs, mathematical definitions, etc. The most common digital representation of mathematical knowledge are papers written in LaTeX language.

Kazan Federal University is working on a project called Lobachevskii-DML, which can be considered a part of the World Digital Mathematical Library (WDML) project [1]-[3]. The WDML project is focused on digitalization and organization of the entire mathematical knowledge in an accessible and efficient way. Within the project, the information is represented through a system of mathematical objects stored in a specially organized repository. This paradigm has seen many implementations so far in the form of local and, in some way, limited DML projects all over the world, e.g., "All-Russian Mathematical Portal Math-Net.RU" [4], "Centre de diffusion de revues académiques mathématiques" [5], "Czech Digital Mathematics Library" [6].

Lobachevskii-DML is a digital mathematical library based on the mathematical knowledge management system OntoMath, which consists of ontologies, textual analytics tools, and applications for mathematical knowledge management [7]-[9]. The semantic search service is an important part of this project; it provides an interface to seek mathematical formulae containing variables that denote predetermined mathematical concepts.

This paper explains the fundamental principles of the markup method used for mathematical documents and the proposed formulae search algorithm [10]. The implementation of the search service based on this algorithm is also discussed, as well as the solution to an efficient formulae search by more than one ontology concept.

The rest of the paper is structured as follows. Section II covers the existing work and compares it with the proposed search method. Section III contains an overview of the Lobachevskii-DML project structure and some details on document processing and ontologies. Section IV outlines the core idea of an original Formula Markup Method. Section V provides details on the accuracy evaluation of the relations established during the document processing. Section VI includes the general description of the search service, as well as of the proposed new features. Section VII focuses on the solution to the implementation of the named features. The results of the search service modification are shown in Section VIII. Section IX covers current results and future possibilities.

II. STATE OF THE ART

Specialized search services that allow seeking information within specific collections of documents, such as scientific articles, is a fast-evolving research area. There are various systems that implement full-text search by keywords and narrow the search to scientific materials. Among these systems are well-known Google Scholar [11] and Microsoft Academic Search [12]. At the same time, a

 ISBN: 978-1-61208-678-1

number of researches focus on implementations that allow making queries in LaTeX markup language: Springer LaTeXSearch [13], (uni)quation [14], EgoMath [15], MIaS [16], Wolfram Formula Search [17]. Some of the mentioned systems implement both approaches, for example, EgoMath allows to formulate search queries alternatively in LaTeX or natural language. However, such systems do not take into account the fact that new and specialized research areas do not often use well-established notations, and different scientific schools may use notations of their own. Lexical terminology is usually more consistent.

The novelty of our approach consists in the integration of the main functionalities of both full-text and formulae search by allowing the user to search for formulae using keywords in natural language. Instead of using mathematical expressions in a query, our approach allows using the textual names of variables that belong to the targeted formula. The search results in a formula and the text fragments which comprise variables and their textual definitions irrespective of their position in the text. Thereby, this approach integrates the functionalities of both full-text search and the search based on seeking a formula by its LaTeX fragments.

III. LOBACHEVSKII-DML

Lobachevskii-DML is built on the digital ecosystem OntoMath [18] which comprises ontologies, textual analytics tools, and applications for mathematical knowledge management. This system consists of the following components:

- Mocassin, an ontology of structural elements of mathematical scholarly papers;
- OntoMath^{PRO}, an ontology of mathematical knowledge concepts;
- Semantic publishing platform;
- Semantic formula search service;
- Recommender system.

The core component of the OntoMath ecosystem (see Figure 1) is its Semantic Publishing Platform. It takes a collection of mathematical articles in LaTeX as an input and builds their semantic representation, which includes metadata, the logical structure of documents, mathematical terminology and formulae.

Mocassin [19], an ontology of structural elements of mathematical papers, is used to identify specific segments, for example, a theorem, a proof, a formula (15 concepts in total). The ontology also defines relations between these segments.

The OntoMath^{PRO} [7][20] concepts are organized into two taxonomies: hierarchy of areas of mathematics, including its sub-fields; hierarchy of mathematical objects such as a set, function, integral, Fourier series, etc. This ontology defines the following relations: taxonomic relation, logical dependency, the associative relation between objects, belongingness of objects to fields of mathematics, the associative relation between problems and methods. The ontology concepts contain labels, definitions, links to external resources and relations to other concepts, as well as formulae and the relevant text fragments describing variables in these formulae. The terminological sources used during the development of the ontology are classical textbooks, online resources like Wikipedia and Cambridge Mathematical Thesaurus, scholarly papers, and personal experience of practicing mathematicians at Kazan Federal University.

Mocassin and OntoMath^{PRO} ontologies are parts of OntoMath ecosystem, however, SALT (Semantically Annotated LaTeX) [21] and AKT Portal (AKTive Portal created by the Advanced Knowledge Technologies research group) [22] are external ontologies.

To be included in Lobachevskii-DML, every article goes through several stages of processing. At the stage of structural markup, the document is annotated with generic structural elements such as titles, paragraphs, sentences, etc. Then, the formulae markup is performed, each expression is classified and some are linked to the text fragments which represent mathematical entities. The instances of the mathematical entities are elicited as the concepts of OntoMath^{PRO} ontology. The semantic search service uses the ontology concepts to provide means for seeking mathematical formulae that contain notations for these concepts.

The Semantic Formula Search is built using the Semantic Publishing Platform and implements an original formula markup method.

IV. FORMULA MARKUP METHOD

In mathematical texts, we identify the following three entities: *mathematical terms, symbolic notations for terms (variables)*, and *mathematical fragments (formulae).* For a mathematical term, we use a Noun Phrase (NP) acting as an extended syntactic model.

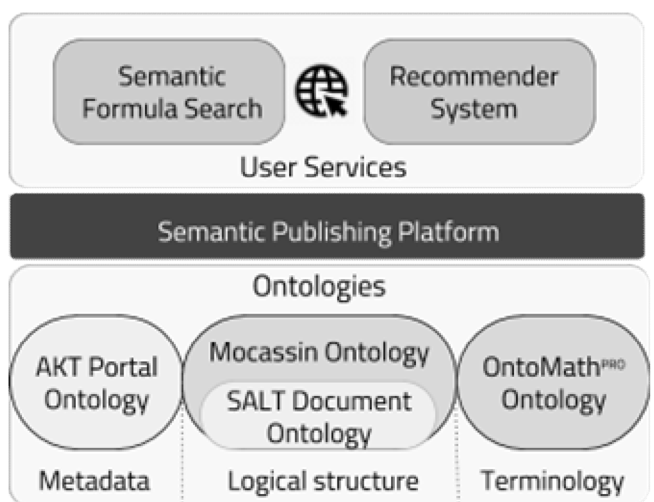

Figure 1. OntoMath ecosystem architecture

Relations among the mentioned entities are defined as follows. The first relation *terms* - *variables* is a textual definition of a symbolic notation through scientific terms within a certain context. The second relation *variables* - *formulae* indicates that a formula contains the symbolic notation. We assume that the appearance of the textual definition of a variable in the neighborhood of its symbolic representation points to a semantic relation between them. The idea of Maximum Permitted Distance (MPD) is used to determine this neighborhood. It is the distance in symbols to the left and to the right of the term which limits the area where a variable can be located. The context of the formula is formed by all the listed entities and relations between them.

The first implementation of the method was a search system for the Russian Wikipedia [23]. The system was based on full-text search within Wikipedia articles containing formulae. That implementation has demonstrated a working efficiency of the method. However, the full-text search focuses on the syntactic features of the searched terms instead of the semantic ones, which leads to a decrease in relevance in the case when the term is a part of some complex term. In order to solve this problem, the second implementation of the method uses preliminary semantic markup.

The formula markup method used in Lobachevskii-DML comprises two steps:

Step 1. Classification of Mathematical Expressions (ME). ME is considered as a *Math*-annotated text. ME consists of symbols for arithmetical and logical operators, variables, variables with index, keywords, and numbers. ME is classified as a *variable* in case it only consists of a variable or a variable with index. Otherwise, it is classified as a *formula*.

Step 2. Establishing relations between variables and formulae. For each *variable*, we search for its occurrences in every *formula* of the document:

Let $\{F\}$ be a set of formulae and $\{P\}$ be a set of variables.

$\forall\ p_i \in P$, if $p_i \in f_k \in F$, the relation $<p_i,f_k>$ is established.

For each relation, the positions of the formula and variables in the text are stored as an attribute. This results in many-to-many relations between formulae and their variables (see Figure 2).

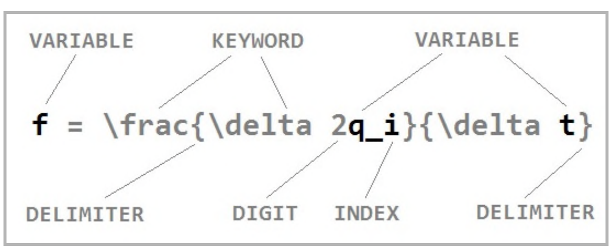

Figure 2. The structure of a mathematical formula.

In Step 1, the *Math*-annotated text is cleared from the service characters of the markup language and excessive space symbols. Next, the fragment is checked against a number of criteria (the length, the number of variables, the presence of relative operators and operations). If the fragment complies with the main criteria, it is considered a formula (a variable or any other type, for example, a table).

When constructing relations between formulae and variables, it is important to pay attention to unique variables in a formula which makes the formula analysis much easier (as opposed to full parsing). Regular expressions are used as a tool for the analysis. First, a formula is split into separate fragments. The delimiters are different types of braces, symbolic notations for arithmetical and logical operations, punctuation characters, spaces, etc. These fragments are then analyzed for belonging to a certain group - keywords (starting with "\"), lower indices (starting with "_"), numbers, etc. If the fragment is not classified at this stage, then it is very likely a single variable. The variables previously found in the text are compared to the variables found in formulae, and at Step 2, the relations of entries of the variables into formulae are established.

V. THE ACCURACY EVALUATION

To assess the accuracy of established relations among mathematical expressions and noun phrases, we used the set of articles from the magazine Izvestiya VUZ Matematika from years 1997-2009. All the articles in Lobachevskii-DML have resulted in 854284 RDF triplets; descriptions of 4190 theorems, 1015 definitions, etc. were included as well. The accuracy was manually evaluated by experts: the relations established with the proposed algorithm were compared to the expected results. The assessment was based on the assumption that the presence of incorrect relations causes irrelevant entries in search results. Besides, the relations that were expected but were never established would not be available for the search service.

Two collections of mathematical documents were processed to evaluate the following parameters:

- correctly related (CR) entities;
- correctly unrelated (CU) entities (which means that there is no NP definition within the context of ME);
- incorrectly related (IR) formulae and NPs (either the NP is semantically irrelevant or a relation was established within unfit context).

The results show that the percentage of correctly processed formulae (*CR*+*CU*) and the percentage of errors (*IR*) vary marginally in response to changing MPD (about 6% for MPD in a range from 15 to 40 symbols). At the same time, the evaluated parameters are changing non-linearly, which means that it is possible to find the most effective MPD for each set of documents. These experiments confirm the stability of the chosen algorithm. For the chosen document collection, the most effective MPD is 20 symbols; the percentage of correct relations is 67.84% (see Figure 3).

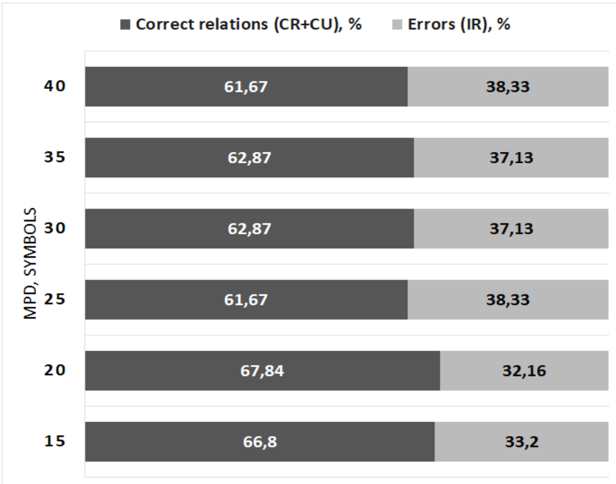

Figure 3. The accuracy of established relations

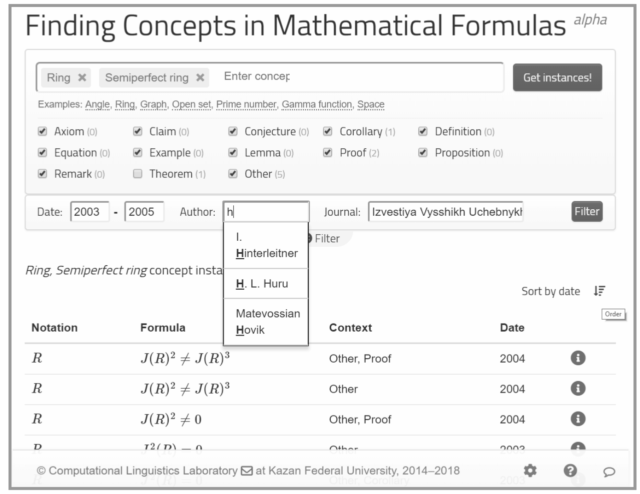

Figure 4. The modified interface of the search service.

VI. SEMANTIC SEARCH SERVICE

After being automatically processed by the formula markup application, the documents become available to the semantic search service [10][24][25] of Lobachevskii-DML system. This implementation is similar to a search by keywords since it does not depend on symbolic notations for a mathematical concept. Keywords are resolved in terms of OntoMath^{PRO} ontology, and the relations of the ontology are used to create a search query. The user is able to limit the search context, for example, search only in definitions or in theorem statements.

The search service is a Web application implemented in JavaScript which uses the RDF query language SPARQL [26] to form a query. The result of the search is represented as a table of contextual data (see Figure 4) which contains a list of data including the symbolic notation of the chosen concept (variable), relevant formula and the context, i.e., the part of the document where the formula was found, as well as the article metadata and its text in PDF.

The alpha version of the search service was released and is now available for testing [27]. Preliminary testing was evaluated by the experienced mathematicians of Kazan Federal University. We have considered their comments and remarks in the modified version of the search service. In particular, the following modifications were suggested:

- search for formulae by more than one ontology concept, which requires dynamically generated queries with many parameters (this feature would allow seeking formulae containing several variables related to the targeted concepts);
- perform additional filtration of the results by metadata of the documents (presetting the range for publication date, the author, the publishers, etc.).

Specific problems and the results of the implementation of these features are explained in the next sections of this paper.

VII. DYNAMIC GENERATION OF SPARQL QUERIES

The implementation of both search by several ontology concepts and the filtration of the results by metadata requires dynamic generation of search queries. A query must be generated in real time in response to user actions given the fact that these actions can be diverse. The SPARQL query language in itself does not support dynamic generation of queries, thus, additional tools are required.

Even though such extension libraries exist in many variations, in case of complex queries with many parameters aiming a specific ontology structure it may be difficult to choose a suitable general-purpose tool. Several approaches to query generation were considered, among which are string concatenation, patterns and macros, and query processors such as Apache Jena ARQ (Automated Request to Query) [28]. In an effort to minimize the cost of time and memory resources for query performance, as well as to easily integrate new features in the existing search service, the decision was made towards string concatenation. This resulted in a simple, yet effective query generation process.

Another important issue that arises when generating queries with many input parameters is query performance optimization. The specifics of the SPARQL language influence the possible approaches to the optimization problem. For example, the modifier DISTINCT is a costly operator, however, SPARQL language provides the modifier FILTER, which allows subqueries usage for filtering conditions and returns a set of unique results. Thus, the usage of the modifier FILTER positively influences the execution speed of a query [29].

The search query implemented in the original version of the search service aimed at one concept only. The query conditions listed all possible relations between a variable and a formula and used the modifier FILTER to select all the relations belonging to the targeted concept. This type of query required a considerable improvement to be able to perform a search by several concepts.

The search service uses endpoint provided by Virtuoso SPARQL Query Service [30]. Virtuoso SPARQL Query Editor [31] was used for query prototyping, testing and debugging.

Figure 5 displays the structure of a search query in SPARQL that determines the connections among concepts, variables, and formulae. The conditions describing relations among formulae and variables take precedence. Then, certain independent conditions are imposed on each variable notation connected to a certain ontology concept (belonging to a class). Then, filters are applied to the class of the variables in such a way so each of the targeted concepts is linked to one of the variables of the formula. This ensures that the keywords are processed as an AND combination. The query also handles some parts of metadata to extract the publication date. The results are grouped by formula.

The execution speed of such a query is decent: 0.8, 1.5 and 2 seconds for one, two and three concepts respectively. The query results are stored in a JavaScript data model that stores formula ID and the document source, its context, variables, LaTeX representation; it also allows querying the complete metadata of the source. Thus, a full-featured base structure was ensured to prepare the data to further processing and display.

Considering the specific nature of the search system, as well as the importance of maintaining the balance between functionality and performance, the search was limited to maximum three concepts per query. This is due to the fact that every targeted variable in the formula has to be linked to an ontology concept. It means that the text containing the targeted formula must include the definition of the variable, and this definition must be recognized during the markup stage, so the relation can be established. As the accuracy of the established relations is influenced by many factors (e.g., comprehensiveness of the ontology, accuracy of noun phrase extraction, the writing style), the increase in the number of the targeted concepts leads to the limited set of formulae that can be potentially found when searching by several concepts. The relation among three variable types is quite sufficient to define the targeted formula in the given collection of the mathematical documents.

```
PREFIX moc: <http://cll.niimm.ksu.ru/ontologies/mocassin#>
SELECT ?formula ?notation0 ?notation1 GROUP_CONCAT(?segment, " ") ...
WHERE  {
     ?formula a moc:Formula;
              moc:hasPart ?notation0, ?notation1; ...
     ?notation0 a ?class0; ...
     ?notation1 a ?class1; ...
     FILTER ( str (?class0) = '...' && str (?class1) = '...')
} GROUP BY ?formula ?notation0 ?notation1 ...
```

Figure 5. The general structure of a search query for two concepts.

VIII. MODIFIED SEARCH SERVICE

The result of this work is an extended version of the Lobachevskii-DML search service (Figure 4). It allows seeking formulae in collections of mathematical documents by one or more ontology concepts. The implementation is based on jQuery [32], a fast and versatile JavaScript library that focuses on the interaction between JavaScript and HTML.

The extended version contains new features such as multiple tag input, a drop-down list of possible inputs containing all the relevant ontology concepts, and a list of possible contexts (which is defined by the Mocassin ontology). An extra panel (hidden by default) contains additional filters: range for publication date, search by a specific author and the publisher.

The implementation of the tag input with drop-down list uses Flexdatalist [33], an autocomplete plugin with multiple input support, so there is no need to perform multiple queries because the data is loaded at the application startup through a single SPARQL query. If the list of the concepts searched has not been changed, the usage of the JavaScript library Knockout [34] allows to avoid performing a new query and hide some of the search results when the user applies the context filter, i.e., only the results found in definitions and proofs are shown.

A search query performed by the service results in a table of contextual data containing the following columns:

- the notation of the variable corresponding to the targeted concept in the particular formula;
- the formula that contains the variable;
- the context of the document in which the formula was found;
- the publication date.

Additionally, the user is able to sort the results by publication date. The search results are grouped by formulae to decrease the redundancy of results. Each result provides an access to further information about the found formula including a list of its linked variables and the metadata of the document containing the formula with a link to the text in PDF (Figure 6).

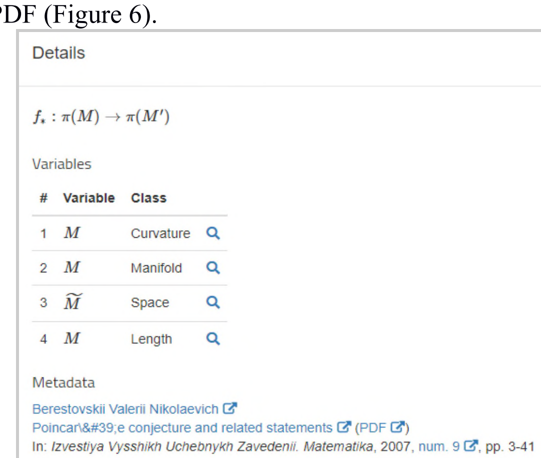

Figure 6. A window with detailed information for the formula.

Furthermore, some minor improvements of the interface were added: a back-to-top button, a search panel fixed to the top of the page that simplifies scrolling through the search results, and concept examples that perform an example search on click.

IX. CONCLUSIONS AND FUTURE WORK

The semantic search service of the Lobachevskii-DML project combines both the convenience of a full-text search and the utility of formulae search. The terms of natural language query are translated into the variables checked for entry in the formulae. Based on the comments received during the preliminary user testing, several new features were implemented in order to make the interface more user-friendly. The challenges implied by the proposed features of the new functionality were solved successfully. As a result, there is an extended version of the Lobachevskii-DML semantic search service with considerable changes in the interface in the sense of both functionality and user experience.

The current version of the search service includes the following features:

- search by more than one concept (up to three), autocomplete for the input boxes, defining the context of the search (structural part of a document);
- access to full metadata on demand, filtration by metadata (date range, author, publisher), sorting by publication date;
- multi-language support (it is possible to search both in English and Russian).

Conducted tests of the extended search service showed stable results while retaining the previously achieved search relevance level (close to 68%). Search queries are generated and executed at a decent speed for the current data set. Preliminary user testing of the search results for one, two and three concepts allow us to conclude that the chosen approach was successful.

Future plans for development include further query generation and performance testing and optimization. It is worth considering the usage of text indexing for querying to make it more efficient. Additionally, the system is easily scalable with more features. More concepts can be available for searching, as the document collection is expanding. For further development, sorting search results by relevance may be considered.

The search service is currently in alpha-testing. The plans for improvement cover user interaction, as well as the algorithm of the system that defines the connections between formulae and ontology concepts and influences search relevance.

It should be noted that under this project we work on the recognition of mathematical documents in PDF which aids the development of the digital library. Handwritten text recognition is not the focus of our research; however, if some tools for digitizing such files and converting them into LaTeX format are available, it will be possible to include the resulting documents in the search service.

ACKNOWLEDGMENTS

This work was funded by the subsidy allocated to Kazan Federal University for the state assignment in the sphere of scientific activities, grant agreement no. 1.2368.2018.

REFERENCES

[1] Developing a 21st century global library for mathematics research, The National Academies Press, 2014.

[2] World Digital Mathematics Library (WDML), https://www.mathunion.org/ceic/library/ world-digital-mathematics-library-wdml, [retrieved: 2018.09.25].

[3] P. J. Olver, “The World Digital Mathematics Library: report of a panel discussion”, Proceedings of the International Congress of Mathematicians, August 13–21, 2014, Seoul, Korea, vol. 1, pp. 773–785. Kyung Moon SA, 2014.

[4] All-Russian Mathematical Portal, http://www.mathnet.ru/, [retrieved: 2018.10.31].

[5] Center for diffusion of academic mathematical journals, http://www.cedram.org/, [retrieved: 2018.10.31].

[6] Czech Digital Mathematics Library, http://dml.cz/, [retrieved: 2018.10.31].

[7] O. Nevzorova, N. Zhiltsov, A. Kirillovich, and E. Lipachev, “OntoMath^{PRO} Ontology: A Linked Data Hub for Mathematics”, Knowledge Engineering and the Semantic Web Communications in Computer and Information Science Volume 468, 2014, pp. 105-119, http://link.springer.com/chapter/10.1007/978-3-319-11716-4_9, arXiv:1407.4833, 2014.

[8] A. Elizarov et al., “Mathematical knowledge representation: semantic models and formalisms”, Lobachevskii J. of Mathematics, 2014, 35, no 4, pp. 347–353.

[9] A. M. Elizarov, A. B. Zhizhchenko, N. G. Zhil’tsov, A. V. Kirillovich, and E. K. Lipachev, “Mathematical knowledge ontologies and recommender system for collections of documents in Physics and Mathematics”, Computer Science, vol. 93, issue 2, pp. 231–233. Springer, Berlin, Heidelberg, 2016.

[10] E. Birialtsev, A. Gusenkov, O. Zhibrik, P. Gusenkova, and Y. Palacheva, “Search in Collections of Mathematical Articles”, Á. Rocha, H. Adeli, L. P. Reis, and S. Costanzo (eds), Trends and Advances in Information Systems and Technologies. WorldCIST'18 2018. Advances in Intelligent Systems and Computing, vol. 745. Springer, Cham, 2018.

[11] Google Scholar, https://scholar.google.com, [retrieved: 2018.09.25].

[12] Microsoft Academic Search, http://academic.research.microsoft.com, [retrieved: 2018.09.25].

[13] The Springer LaTeX Search, http://latexsearch.com, [retrieved: 2018.09.25].

[14] (Uni)quation. Math expression search engine, http://uniquation.com, [retrieved: 2018.01.07].

[15] J. Misutka and L. Galambos, “Extending Full Text Search Engine for Mathematical Content”, Proceedings of DML, pp. 55–67, 2008.

[16] P. Sojka and M. Líška, “Indexing and Searching Mathematics in Digital Libraries”, J. H. Davenport, W. M. Farmer, J. Urban, and F. Rabe (eds), Intelligent Computer Mathematics. CICM 2011. Lecture Notes in Computer Science, vol 6824. Springer, Berlin, Heidelberg

[17] The Wolfram Functions Site, http://functions.wolfram.com, [retrieved: 2018. 09.25].

[18] A. Elizarov, A. Kirillovich, E. Lipachev, and O. Nevzorova, Digital Ecosystem OntoMath: Mathematical Knowledge Analytics and Management. In: L. Kalinichenko, S. Kuznetsov, and Y. Manolopoulos (eds) Data Analytics and Management in Data Intensive Domains. DAMDID/RCDL 2016. Communications in Computer and Information Science, vol 706. Springer, Cham

[19] V. Solovyev and N. Zhiltsov, “Logical structure analysis of scientic publications in mathematics”, Proceedings of the International Conference on Web Intelligence, Mining and Semantics (WIMS'11), 2011, pp. 21:1–21:9, ACM, 2011.

[20] E. V. Birialtsev et al., “Methods for Analyzing Semantic Data of Electronic Collections in Mathematics”, Automatic Documentation and Mathematical Linguistics, 2014, vol. 48, no. 2, pp. 81–85, 2014.

[21] T. Groza, S. Handschuh, K. Möller, and S. Decker, SALT - Semantically Annotated \LaTeX for Scientific Publications. In: Franconi E., Kifer M., May W. (eds) The Semantic Web: Research and Applications. ESWC 2007. Lecture Notes in Computer Science, vol. 4519. Springer, Berlin, Heidelberg

[22] The AKT Reference Ontology, http://projects.kmi.open.ac.uk/akt/ref-onto/, [retrieved: 2018.09.25]

[23] E. V. Birialtsev, A. M. Gusenkov, and O. N. Zhibrik, “Some approaches to markup of scientific texts containing mathematical expressions”, Uchenye Zapiski Kazanskogo Universiteta. Seriya Fiziko-Matematicheskie Nauki, 2014, vol. 156, no 4, pp. 133–148, 2014.

[24] A. Elizarov, A. Kirilovich, E. Lipachev, and O. Nevzorova, “Mathematical Knowledge Management: Ontological Models and Digital Technology”, CEUR Workshop Proceedings, 2016, vol. 1752, pp. 44-50, http://ceur-ws.org/Vol-1752/paper08.pdf, 2016.

[25] Finding Concepts in Mathematical Formulas alpha, http://lobachevskii-dml.ru:8890/mathsearch, [retrieved: 2018.09.25].

[26] SPARQL Query Language for RDF, v. 1.1, The W3C SPARQL Working Group, https://www.w3.org/TR/sparql11-overview/, [retrieved: 2018.09.25].

[27] Lobachevskii-DML search service, http://lobachevskii-dml.ru:8890/mathsearch, [retrieved: 2018.10.31].

[28] ARQ - A SPARQL Processor for Jena, https://jena.apache.org/documentation/query/, [retrieved: 2018.09.25].

[29] Bob DuCharme, “Learning SPARQL. Querying and Updating with SPARQL 1.1”, O'Reilly Media, 2013.

[30] Virtuoso SPARQL Query Service, http://vos.openlinksw.com/owiki/wiki/VOS/VOSSparqlProtocolVirtuoso, [retrieved: 2018.09.25].

[31] Virtuoso SPARQL Query Editor, http://lobachevskii-dml.ru:8890/sparql, [retrieved: 2018.09.25].

[32] jQuery, a JavaScript library, https://jquery.com/, [retrieved: 2018.10.31].

[33] Flexdatalist - jQuery autocomplete/datalist, http://projects.sergiodinislopes.pt/flexdatalist/, [retrieved: 2018.09.25].

[34] Knockout, http://knockoutjs.com/, [retrieved: 2018.09.25].