Semantic Search System with Metagraph Knowledge Base and Natural Language Processing

Valery Terekhov, Anton Kanev
Bauman Moscow State Technical University
Moscow, Russia
terekchow@bmstu.ru, aikanev@bmstu.ru

***Abstract*—Currently, various investigations are actively carried out to improve the precision and recall of information retrieval. Many authors associate this process with the need to analyze the meaning of words. The authors of this paper have proposed a semantic search method using natural language processing and the metagraph knowledge base. The general model and main algorithms of the proposed method for indexing and information extraction are described. Natural language processing capabilities affect the amount of data available for search, thus, the recall of the information extraction system was measured. Marking up a dataset according to meaning depends on the situation and is subjective. Therefore, the precision of semantic search was assessed on an unlabeled dataset using the methodology proposed by the authors. To increase recall, semantic search is complemented by keyword search, and semantics results are used to change the ranking of user query results. The authors suggested set of queries for this investigation. The ranking order for semantic and regular keyword searches was estimated using the metric proposed by the authors.**

I. Introduction

A. Information retrieval

Information retrieval is the process of identifying in a set of documents that meet the specified conditions or contain the necessary data [1]. Investigation in this area began in the 1950s [2]. Two main directions exist: full text search and metadata search. In the first case the search is carried out throughout the entire text document. And in the second case it is preformed according to the selected data: topic, keywords, etc. There is a division of information retrieval into statistical, machine learning, processing of semantic networks, combined.

It should be also noted the application of text mining for information retrieval. Large companies use large commercial information extraction projects, such as Abbyy Compreno [3] based on deep natural language processing to accurately find information in documents. But they turn out to be slow and expensive to implement and require the involvement of a large number of employees. Therefore, application of these systems is limited to internal document repositories of companies.

Google applied in their popular search engine the modern neural architecture BERT since 2019. It works with various languages and allows to interpret better user queries. It improves quality of approximately 10% of queries that are long, colloquial or include prepositions. This architecture is associated with the need to use high computing power. Therefore, specifically for BERT the company launched a new powerful TPU (Tensor Processing Unit) cloud.

One of the problems of information retrieval is the resolution of lexical disambiguation WSD (Word sense disambiguation). It requires to choose an appropriate meaning for a polysemantic phrase, depending on the context. Various methods are used to solve this problem [4], [5].

B. Semantic search

Semantic search is a type of information retrieval that uses the semantic meaning of words and expressions and analyzes the context for a more accurate interpretation of the user query and improving search results [5], [6], [7]. It is primarily associated with knowledge processing. An example of semantic search is synonym processing.

Text mining is often used for semantic search [8]. At the same time, other methods can be used. For example, the interpretation of a query from keywords into a query to a database [9], the use of a specific query language for a specific purpose [10], the use of basic knowledge in the form of thesaurus [11]. Semantic search is used in various fields: medicine [8], e-mail management [9], search for necessary web services [12], search in XML documents [10], cultural heritage objects [11].

Semantic search is also understood as the use of specialized semantic query languages [13], for example, SPARQL. Therefore, works of individual researchers are directed at translating queries from keywords into the language of semantic queries to ontology [13], [14]. In other studies [15], an ontology is used to interpret a user query by associating keywords with ontology objects.

In the study [15], the Lucene library is used for indexing and morphological analysis during the process of searching for concepts in ontology. Some researchers [6, 11, 14] associate semantic search with the semantic web. The authors also oppose their approach to the semantic web, because they focus on the need to automatically extract data for ontology from the processing texts.

C. Vector space model

A search query can include many words, and not all of them may appear in the searched documents. Therefore, search

results usually contain several documents sorted according to ranking. One of the more well-known ranking functions is TF-IDF (Term Frequency Inverse Document Frequency). The TF-IDF measure indicates the importance of the word for the specified document. It is often used in text mining tasks such as information retrieval and text classification.

The TF-IDF metric (1) is based on the use of a vector space model, where each document and query corresponds to a vector in a multidimensional space with number of dimensions equal to the number of unique words in all documents. It includes two other metrics TF (2) and IDF (3).

$$TF\text{-}IDF(t,d,D) = TF(t,d) \cdot IDF(d,D) \qquad (1)$$

$$TF(t,d) = \frac{n_t}{n_d} \qquad (2)$$

$$IDF(t,D) = \log \frac{N}{n(t)} \qquad (3)$$

where t is a word, d is a document from a collection of documents D, n_t is the number of times a word t is used in a document d, n_d is the number of words in a document d, N is the number of documents in D, $n(t)$ is the number of documents in D that contain the word t.

Semantic search includes not only vector space model but also other methods. Some authors use ordinary TF-IDF metric in their research. While in study [10] different metrics for ranking of semantic search results were combined using advantages of both keyword and semantic search.

D. Search engines

Search engine is a computer system for finding information. The basis of the information retrieval system is a search engine. Elasticsearch is a freely replicable and one of the most popular search engines. It is based on the use of the Lucene library. Lucene is a free high-performance full-text search library from the Apache Foundation.

Microsoft Azure is a cloud platform providing full text search engine [16] that includes cognitive search to improve quality of results and uses artificial intelligence techniques for this purpose. It is based on architecture and model of Lucene library. This search engine also allows to maintain both the simple query language and the Lucene extended query language. The first one is more commonly used and avoids logic of query. The second one includes modifiers for specific types of queries with extended information such as fuzzy, wildcard search, regular expressions, boolean operators, etc.

Azure also includes lexical analysis of queries. Lexical or morphological analysis only applies to search by term or phrase. It cannot be applied to fuzzy search queries, regular expressions, wildcard searches. In this case, this type of query is added immediately to the query tree, bypassing the stage of lexical analysis. Azure Cognitive Search supports a wide variety of analyzers for Lucene and Microsoft. After analysis comes the stage of retrieving a collection of documents, followed by their evaluation to display them in a specific order.

In Azure, it is possible to use artificial intelligence to enrich the indexing. It uses image processing and natural language processing. Natural language processing includes entity recognition, language recognition, key phrase extraction, and sentiment analysis [16]. The document representation obtained with a standard AI pipeline is removed after indexing. The knowledge store is used when AI-enriched documents need to be saved.

The technology of skillsets is used in Azure during artificial intelligence processes. A skillset is a collection of skills that are used consistently in a specific pipeline. Skills can be built-in based on Cognitive Services, or the user can define his own skill and add it to the processing pipeline. The Text Analytics API, a part of Cognitive Services, provides natural language processing and it is a cloud-based service.

II. METHODOLOGY

Previously, the authors of the work proposed a technique for extracting information using natural language processing to build an ontology. In the current work, it was taken as a basis for the creation of a semantic search system. The ontology is presented using a metagraph model, and it is proposed to use a semantic index of concepts and a modified TF-IDF metric to search and rank results instead of keywords.

A. General model of semantic search

Earlier in work [17], the authors proposed a method for extracting information from text for a knowledge base. The method successfully mines concepts and connects them with different types of relations. In the new research, this method is taken as the basis for integration with a search engine.

Each concept has its extension and intension. The extension is a set of child concepts and represents different cases of the concept. The intension is a set of parent concepts. It is the meaning of concept. Each word or whole phase from text connected with its concept can be analyzed using meaning.

For representation of concepts and their relations the metagraph model was chosen which is a type of complex network [18]. Processing of this model is described at [19]. Metagraph knowledge representation allows to extend flat semantic network with emergence feature [20, 21]. It allows to describe each concept or relation with its own subgraph.

The authors introduce the following designations (4) and (5) to describe semantic search process.

$$m(phrase) = concept \qquad (4)$$

$$s(concept) = \{< concept_i, k_i >\} \qquad (5)$$

where $m(phrase)$ is a function that assigns a certain $concept$ to a sequence of words from a query or document $phrase$,

$s(concept)$ is a function that returns a set of related concepts $concept_i$ for a specified $concept$, and weights k_i of respective relations characterizing the degree of correspondence with the original concept from knowledge base.

The system analyzes the indexed documents using a function $m(phrase)$ to carry out semantic search, finds a set of related concepts in $s(concept)$ and create a semantic index based on them. Then functions $m(phrase)$ and $s(concept)$ are calculated to the search query. The last step includes concept-based search index that is applied to the specified set of concepts from the query.

B. Semantic index

The search index based on concepts is presented in the following form (6).

$$Index = \{< concept, \{i_{d_1}, .., i_{d_n}\} >\} \quad (6)$$

where a $concept$ instead of a word is assigned with identifiers of documents $i_{d_1}, .., i_{d_n}$.

A modified measure TF-IDF is proposed for a concept-based search index through its constituent metrics TF (7) and IDF (8).

$$IDF'(concept) = \log \frac{N}{n(concept)} \quad (7)$$

$$TF'(concept, d) = \frac{n_{concept}}{n_d} \quad (8)$$

where N is the number of documents, $n(concept)$ is the number of documents containing the $concept$, $n_{concept}$ is the number of references on $concept$ in the document d, and n_d is the total number of concepts in the document d.

C. Information extraction

It is necessary to obtain concepts and relation for knowledge base before semantic search. Therefore, a method of natural language processing for extracting information from text was included into the model. The input of this method is natural language text, that is divided into sentences, that in turn are divided into words. Let's represent it as a two-dimensional array of words for each text document (9).

$$Text = [sentence] = [[word]], Text \in T \quad (9)$$

where T is a set of natural language texts.

Indexed documents and search queries are texts. For a given search query during the process of information retrieval it is necessary to find a set of documents and arrange them in accordance with the ranking results. It is required to associate each search query with its own ordered set of documents, which is presented in (10) and (11).

$$I(query) = [document_i] \quad (10)$$

$$query \in Q, document_i \in D, Q \cup D \subset T \quad (11)$$

where D is a set of documents, Q is a set of search queries, I is a mapping specifying an information retrieval. The semantic search can be specified as follows through (12) and (13).

$$I_s(query) = S(A(query)) \quad (12)$$

$$A(query) = \{C_q, R_q\}, C_q \subset C, R_q \subset R \quad (13)$$

where I_s is the mapping for semantic search, A is the mapping corresponding to the natural language processing, S is the mapping using the knowledge base to obtain the required documents, C is the set of concepts, and R is the set of relations in the knowledge base.

D. Design

The general schema of the proposed method presented on Fig. 1. It includes three main steps for both document and query: natural language processing pipeline, knowledge base and sematic index. Natural language processing consists of tokenization, morphological analysis, syntactic analysis, context analyzer and application of semantic rules.

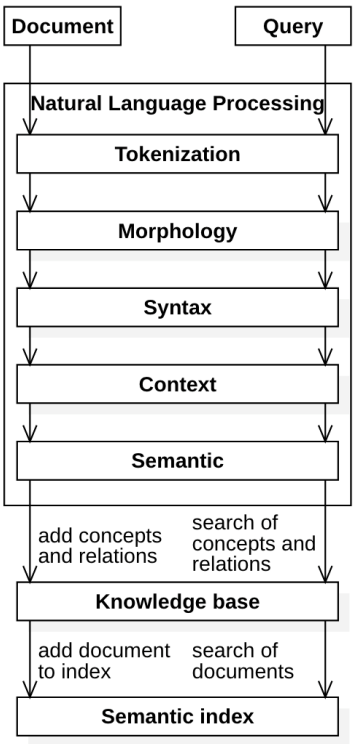

Fig. 1. General schema for semantic search method

The authors of the paper developed the Semantic Search module (Fig. 2) to implement an information retrieval system, consisting of three packages for the implementation of the knowledge base (knowledge), information search (search) and natural language processing (analyzer). In addition to these three packages, a separate evaluation package was also developed to evaluate the precision, recall and ranking of query results for the proposed method.

Knowledge package implements methods to get and to put concepts (class Entity) and relations (class Relation) into the knowledge base. Search package contains SemanticIndex class for document indexing and result ranking. The weights of the relations in this study are taken equal to one, and for their calculation in the future it is planned to use machine learning based on the distribution of concept usage in various texts

The analyzer package for natural language processing implements morphological and syntactic analysis. Text class is designed for tokenization. RussianMorphology library is used for morphological analysis of English and Russian texts. Since the texts are considered in different languages, two different modules english and russian are used for morphological analysis. The analyzer package provides an interface for interacting with these modules and processing the received data about the initial form of the word, grammatical categories and parts of speech.

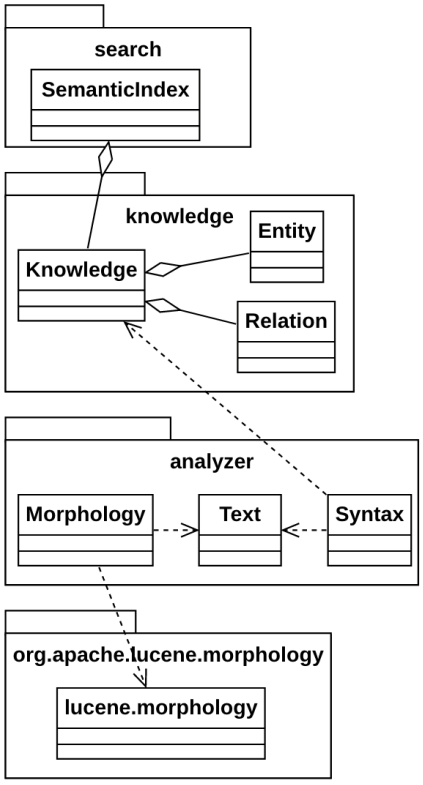

Fig. 2. Class diagram of developed system

Syntax analysis is based on constituent grammar and it creates parse tree for each sentence. With methods of Syntax class tree nodes are extended with concepts and relations by knowledge package. For processing of pronouns, the package includes context analyzer.

III. Evaluation of precision and recall

To assess the possibilities of semantic search, a methodology for calculating precision, recall and ranking is described, which allows using unlabeled datasets. The indicators calculated during the study confirmed the improvement in search results. To study all metrics in this work, the OpenCorpora dataset was used. As the initial data the results of the Lycene library were compared with semantic search using the same morphological analyzer.

Due to the limited number of used parsing rules, it was possible to increase only the precision due to semantic search, and the recall turned out to be lower. Therefore, a combination of two types of search by concepts and by words was used, which increases the recall, and also maintains a high value of precision. This saved the number of documents, and the advantage of semantic search is used to reorder the search results. Ranking is evaluated at the second part of the investigation.

A. Precision and recall

The precision and recall of the developed system were assessed during investigation. Various estimates are used to assess the quality of information retrieval [1]. The search precision is determined by (14).

$$Precision = \frac{\left|D_{rel} \cap D_{retr}\right|}{\left|D_{retr}\right|} = \frac{N_{correct}}{N_{find}} \qquad (14)$$

where D_{rel} is the set of relevant documents in the database, D_{retr} is the set of documents found by the system, $N_{correct}$ is the number of correctly identified documents, and N_{find} is the total number of documents found by the system.

Precision shows the number of correctly classified documents in relation to the number of documents found by the system. Precision can be high if the system detects only few correct documents. Therefore, the second characteristic of recall exists (15).

$$Recall = \frac{\left|D_{rel} \cap D_{retr}\right|}{\left|D_{rel}\right|} = \frac{N_{correct}}{N_{true}} \qquad (15)$$

where N_{true} is the number of documents that are actually related to the search query. Recall indicates the ratio of correctly found documents in relation to the number of relevant documents, that should have been identified by the system.

But recall can be high if the system produces a large number of documents, even with incorrectly classified ones. A measure that combines these two metrics is called the F-measure or the Van Riesbergen measure. Balanced or F_1-measure is the most commonly used (16).

$$F_1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \quad (16)$$

B. Precision of semantic search system

Precision and recall assessing during analysis of meaning is not a trivial task. In addition to the need to manually mark up the data for comparison, the markup can differ from person to person. Therefore, the authors of the paper tried to find criteria that do not require manual data markup and do not depend on who carried out this markup.

For this purpose, recall was assessed by the number of analyzed words during the search of semantic concepts compared to the word-based search (Table I). This recall is influenced by the results of morphological analysis through the number of words having uniquely determined form.

The parsing rules are also affected, because there are a limited number of them and they do not allow to parse all the words from the text. But indexing of other parts of speech as words without correlating them with other concepts and relations in the knowledge base can significantly improve search recall.

Precision is influenced by many parameters. These are the possibilities of morphological analysis, which may not always work correctly. There can also be exceptions during applying parsing rules or mistakes, especially when parsing grammatically incorrect sentences.

It is difficult to assess restrictions of NLP pipeline. Therefore, for each concept in the knowledge base the number of documents (Table I) that were found using semantic search and the number of documents that the system returned without applying semantic search based on the joint use of words in text document were calculated to get the search precision in comparison with the Lucene library.

TABLE I. RECALL OF WORD ANALYSIS AND PRECISION OF SEARCHING BY THE NUMBER OF DOCUMENTS

Parameter	Total words and concepts	Nouns, adjectives and complex concepts	Complex concepts
Recall (%)	100	45.4	45.4
Recall (number of words)	1545898	701924	701924
Number of concepts/words	187396	165629	114852
Average number of rejected documents for each concept after semantic search (%)	31.4	35.4	51.3

A new function was developed to evaluate these values. It receives a knowledge base data obtained from texts as input. Then it looks at the list of all concepts passed to it. A second function returns a list of the most common ancestors for a given concept. This function scans all outgoing relationships of all types for a concept.

The result is the uppermost concepts that have no ancestors, but are expressed in the text in the form of words. For each parent concept, a list of documents is calculated in which it is used, and then the intersection of the sets of these documents is found for all parent concepts.

The obtained data is used to calculate the average share of documents that turned out to be redundant based on the results of comparing the two types of search. The denominator in this fraction is the number of documents found in word searches. These are documents that contain words that are related to the required concept, but that are used separately in the text.

C. Evaluation of TF-IDF metric

To evaluate the quality of the information extraction and the semantic search system, a study was carried out on the assessment of the change in TF and IDF metrics when using the proposed semantic search on the OpenCorpora text set (Table II). For each word for each text, the difference between the TF metric, obtained using the conventional Lucene search engine, and the semantic search system was calculated.

TABLE II. PRECISION OF SEARCH BY TF-IDF

Parameter	Total words and concepts	Nouns, adjectives and complex concepts	Complex concepts
Average change of TF for each concept (%)	37.6	42.6	61.4
Average change of IDF for each concept (%)	35.5	40.2	58.0
Average change of TF-IDF for each concept (%)	38.3	42.9	62.6

The authors evaluated the change in the percentage of the TF, IDF and TF-IDF metrics, taking into account the fact that some of the documents turned out to be superfluous, and some of the use of words in correctly found documents may turn out to be unnecessary. The results of this study of the recall and precision of search by the number of documents and the TF-IDF metric are also shown in Fig. 3.

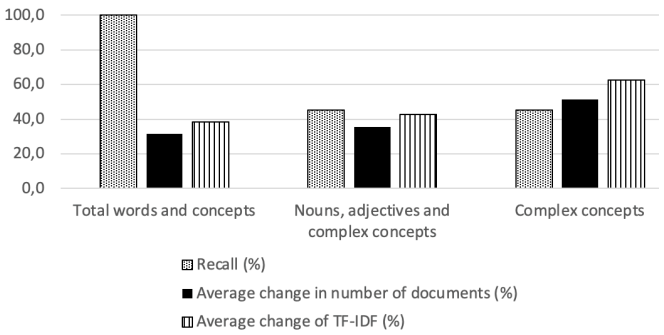

Fig. 3. Recall of word analysis and search precision

TF is summarized for search results of each word, and then the results are summed across all documents. The IDF from the resulting collection is simply summarized. TF-IDF is calculated

similarly to TF, but during the summation, the results are multiplied by the corresponding IDF value for the specified concept. For all three metrics (17), word-based search data is used as the denominator.

$$P_{tf} = \sum_{doc} \left(\frac{\left| tf_c - \sum_t tf_t \right|}{\sum_t tf_t} \right) \cdot 100\% \qquad (17)$$

$$P_{idf} = \frac{\left| idf_c - \sum_t idf_t \right|}{\sum_t idf_t} \cdot 100\%$$

$$P_{tfidf} = \sum_{doc} \frac{\left| tf_c \cdot idf_c - \sum_t tf_t \cdot idf_t \right|}{\sum_t tf_t \cdot idf_t} \cdot 100\%$$

where P_{tf}, P_{idf} and P_{tfidf} is precision for TF, IDF and TF-IDF, c is a concept, and t are the words of the concept c.

In semantic search, the metric is calculated separately for each concept. During search by words, the TF-IDF metric is added for a compound concept. The difference in the IDF metric was calculated for each word. If a concept consists of one word, the values of the TF and IDF metrics are the same. If a concept consists of several words, then they are different.

IV. Analysis of ranking

The proposed technique improves the precision and recall of the search. But a real system does not contain all rules for parsing sentences and extracting all the information for the knowledge base. Some words may not be concerned as concepts in semantic analysis.

Therefore, changes were made to overcome the problem with the analysis of verbs, adverbs, those parts of speech that are not analyzed by the current set of rules. All these words are added to the knowledge base as new concepts and indexed during the processing of text documents. The only exceptions are words for that it is not possible to determine unambiguously the initial form.

But adding words in search results decreases search precision. Therefore, semantic search results are used to change the ranking order of query results. The metric d_d proposed by the authors and the Levenshtein distance were used to assess the ranking. Moreover, the values of these metrics were normalized in each case to the total number of different documents in the results of the two systems.

There are issues during evaluating a list of documents: documents cannot be repeated in the results of one system and documents can be located in different places in the list of results, including positions that are far from each other. Levenshtein distance does not take into account transpositions of sequence elements. Therefore, another metric was proposed to assess the ranking of the results of the two search systems. It was decided to use a different estimation taking into account the proximity, that is equal to the sum of the estimates for each unique element from the combination of both sequences (18).

$$d_d = \sum_{c \in A \cup B} d_d(c) \qquad (18)$$

$$d_d(c) = \begin{cases} 1, (c \notin A) \cup (c \notin B) \\ \dfrac{div}{length}, (A_i = B_j = c) \end{cases}$$

$$div = |i - j|$$

$$length = |A \cup B|$$

where div is the difference in the indices of the element in two sequences, $length$ is the number of unique elements in the two sequences.

The score for one element is 1 if a character needs to be deleted or inserted, or $\frac{div}{length}$ if the element is contained in both sequences. This metric matches the Levenshtein distance if the sequences do not contain common elements.

The system was tested on the developed set of 130 queries. It includes eleven groups that can be combined into two large blocks: word queries and queries with modifiers. Measurements of metrics carried out during experiment was averaged over groups. Research was conducted with Lucene (Table III). As a result, the number of documents, found in the process of semantic search, approached the data obtained from Lucene using the RussianMorphology library. This is especially noticeable on a group of "long queries".

It can be seen from the results that for the first block, precision and recall has high values, while the distance is small. At the same time, recall and precision reach maximum values, and the ranking order changes more than 15 percent. For the second block, the situation is the opposite. Queries with modifiers need to use a separate query sintax parsing as well as an optional index. It can be accomplished by combining the benefits of a semantic index and a specialized index for modifiers.

TABLE III. Results of precision, recall and ranking for different types of queries

Query type	Precision	Recall	F_1	Levenshtein	Distance
Single word queries	0,60	0,53	0,56	0,44	0,29
Short queries	1,00	0,86	0,92	0,73	0,24
Short queries with prepositions	1,00	0,92	0,96	0,98	0,23
Long queries	1,00	0,99	0,99	0,98	0,15
Queries with verbs and adverbs	0,67	0,66	0,66	0,45	0,22

Query type	Precision	Recall	F_1	Levenshtein	Distance
Quoted phrases	0,01	0,20	0,02	0,79	0,80
Grouping queries	0,00	0,00	0,00	1,00	1,00
Boolean operators	0,44	0,48	0,46	0,79	0,61
Wildcard queries	0,10	0,10	0,10	0,29	0,22
Fuzzy queries	0,10	0,09	0,09	0,68	0,62
Proximity Search	0,05	0,20	0,08	0,95	0,97

The metric d_d value has decreased significantly, which began to differ greatly from the Levenshtein distance for non-specialty query groups. On the groups of queries "long queries" and "short queries with prepositions" it can be seen that the Levenshtein distance takes mostly maximum values, while d_d has a very small value.

V. Conclusion

In the paper the authors described a semantic search technique that uses natural language processing and the metagraph knowledge base for information retrieval. The results of the paper have demonstrated the possibilities of using metagraph knowledge base and semantic concept index to increase the quality of information retrieval through analysis of meaning. The search precision was assessed and confirmed its improvement for the case of semantic search. The combination of two types of search in the analysis of concepts and words made it possible to analyze all words from texts, as well as to increase the search precision by 30% compared to the search for keywords. Proposed metric for evaluation of ranking results shows a significant change in their order compared to keyword search. The developed system made it possible to improve the ranking of search results by an average of 15% for queries without modifiers according to the metric proposed by the authors. Therefore, future research includes the use of more complete NLP analyzers, including new languages, as well as investigation of machine learning for calculating relation weights on inference in the knowledge base and results of semantic search.

References

[1] K. van Rijsbergen. Information retrieval. London: Butterworths. 1979.

[2] C. Mooers, "Information retrieval viewed as temporal signaling", *in Proc. of the International Congress of Mathematicians*, 1950, vol. 1, pp. 572–573.

[3] E. Manicheva, M. Petrova, E. Kozlova, and T. Popova, "The Compreno Semantic Model as an Integral Framework for a Multilingual Lexical Database", *in Proc. 24th International Conference on Computational Linguistics, the 3rd Workshop on Cognitive Aspects of the Lexicon (CogALex-III)*, 2012, pp. 215-230.

[4] M. Sussna, "Word sense disambiguation for free-text indexing Using a Massive Semantic Network", *in Proc. of the second international conference on Information and knowledge management*, 1993, pp 67-74.

[5] N. Senthil Kumar and M. Dinakaran, "An algorithmic approach to rank the disambiguous entities in Twitter streams for effective semantic search operations", *Sadhana*, 2020.

[6] R. Guha, R. McCool, and E. Miller, "Semantic Search", *in Proc. of the 12th international conference on World Wide Web*, 2003.

[7] J. Zhong, H. Zhu, J. Li, and Y. Yu, "Conceptual Graph Matching for Semantic Search", *in Proc. Conceptual Structures: Integration and Interfaces*, 2002, pp. 92-106.

[8] P. Thomas, J. Starlinger, A. Vowinkel, S. Arzt, and U. Leser, "GeneView: a comprehensive semantic search engine for PubMed", *in Proc. Nucleic Acids Research*, vol. 40, 2012, pp. 585–591.

[9] E. Kandogan, R. Krishnamurthy, S. Raghavan, S. Vaithyanathan, and H. Zhu, "Avatar Semantic Search: A Database Approach to Information Retrieval", *in Proc. of the ACM SIGMOD International Conference on Management of Data*, 2006.

[10] S. Cohen, J. Mamou, Y. Kanza, and Y. Sagiv, "XSEarch: A Semantic Search Engine for XML", *in Proc. 2003 VLDB Conference*, 2003, pp. 45-56.

[11] G. Schreibera, A. Aminb, L. Aroyoa, M. Van Assema, V. De Boerc, L. Hardmanb, M. Hildebrandb, B. Omelayenkoa, J. Van Osenbruggenb, A. Tordaia, J. Wielemakerc, and B. Wielinga, "Semantic annotation and search of cultural-heritage collections: The MultimediaN E-Culture demonstrator", *SSRN Electronic Journal*, 2008.

[12] N. Srinivasan, M. Paolucci, and K. Sycara, "An Efficient Algorithm for OWL-S Based Semantic Search in UDDI", *in Proc. International Workshop on Semantic Web Services and Web Process Composition*, 2004, pp. 96-110.

[13] Q. Zhou, C. Wang, M. Xiong, H. Wang, and Y. Yu, "SPARK: Adapting Keyword Query to Semantic Search", *in Proc. International Semantic Web Conference Asian Semantic Web Conference (ISWC)*, 2007, pp. 694-707.

[14] Y. Lei, V. Uren, and E. Motta, "SemSearch: A Search Engine for the Semantic Web", in Proc. Managing Knowledge in a World of Networks, 2004, pp. 238-245.

[15] T. Tran, P. Cimiano, S. Rudolph, and R. Studer, "Ontology-Based Interpretation of Keywords for Semantic Search", *in Proc. International Semantic Web Conference Asian Semantic Web Conference (ISWC)*, 2007, pp. 523-536.

[16] Microsoft official website, Azure Cognitive Search, Web: https://docs.microsoft.com/ru-ru/azure/search/search-lucene-query-architecture.

[17] A. Kanev, S. Cunningham, and V. Terekhov, "Application of formal grammar in text mining and construction of an ontology", *in Proc. Internet Technologies and Applications (ITA)*, 2017, pp. 53-57.

[18] V.M. Chernenkiy, Yu.E. Gapanyuk, G.I. Revunkov, Yu.T. Kaganov, Yu.S. Fedorenko, and S.V. Minakova, "Using metagraph approach for complex domains description," *in Selected Papers of the XIX International Conference on Data Analytics and Management in Data Intensive Domains (DAMDID/RCDL 2017)*, Oct. 2017, pp. 342-349.

[19] V.M. Chernenkiy, Yu.E. Gapanyuk, A.N. Nardid, and N.D. Todosiev, "The Implementation of Metagraph Agents Based on Functional Reactive Programming," *in Proc. 26th Conference of Open Innovations Association (FRUCT)*, Yaroslavl, Russia, 2020, pp. 1-8, doi: 10.23919/FRUCT48808.2020.9087470.

[20] V. Chernenkiy, Y. Gapanyuk, A. Nardid, M. Skvortsova, A. Gushcha, Y. Fedorenko, and Picking R, "Using the metagraph approach for addressing RDF knowledge representation limitations," *in Proc. Internet Technologies and Applications (ITA)*, 2017, pp. 47-52.

[21] V.M. Chernenkiy, Yu.E. Gapanyuk, A.N. Nardid, A.V. Gushcha, and Yu.S. Fedorenko, "The Hybrid Multidimensional-Ontological Data Model Based on Metagraph Approach," *in Proc. Perspectives of System Informatics*, 2018, pp. 72–87.