Evaluation issues of query result ranking for semantic search

A I Kanev[1] **and V I Terekhov**[1]

[1] Computer Science and Control systems department, Bauman Moscow State Technical University, Moscow, 105005, Russia

aikanev@bmstu.ru

Abstract. Application of semantic in information retrieval is a dynamically developing area. Nowadays, elements of semantic search are used in popular systems such as Microsoft Azure, Abbyy Intelligent Search, Google Search with BERT. Using sematic search, it is possible to obtain documents that contain exact meaning instead of set of words. But Lucene is still one of the most popular libraries for search purpose and it has its own syntax for fuzzy, wildcard, proximity and other modifiers for queries. To evaluate precision and recall of search the authors have created a list of queries and divided it into groups according to a query type. The article contains results of this investigation for semantic search with metagraph knowledge base in comparison to Lucene with the same morphological analyzer. The quantity of documents for two types of search may be the same but ranking should be different. Ranking of queries is another issue and its evaluation is not a trivial task. In this article the authors applied Levenstein distance but then proposed a new method for comparison of ranking given by different search engines. All results were obtained on Open Corpora text corpus.

1. Introduction

Information search is constantly improving, and as Google emphasizes, there are still many unsolved problems [1]. To improve search results, it is necessary to take into account the meaning of the words [2, 3]. Natural processing is often used for this purpose, which includes two large areas: syntactic-semantic analyzers and machine learning. Large companies use commercial information extraction projects for accurately search such as Abbyy Intelligent Search [3], based on sophisticated syntactic and semantic analyzers. But they turn out to be expensive to implement.

Since the end of 2019, Google has been using the BERT neural network in various languages, including Russian, to improve search queries [1]. BERT is used to analyze only 10% of queries: long, colloquial, or containing prepositions. It is an improvement in the interpretation of the query itself, and not an analysis of the entire collection of documents. The use of BERT is associated with complex models and requires significant computing power. The company began to use a new Cloud TPUs (Tensor Processing Unit).

Various knowledge representations are also used to improve the search [2, 3, 4, 5]. RDF model is widely used for this purpose [6]. Metagraphs are one of the actively developing areas. The paper [7] describes the problems that arise when using RDF, which can be solved using the metagraph emergence property. Previously, the authors in [8] described a method for extracting information from a text for a semantic network using natural language processing. This method was used to modify the metagraph knowledge base in a semantic search system.

One of the most popular libraries for information search is Lucene, which is part of the Solr and Elastic search engines [9]. Lucene is a free high-performance full-text search library from the Apache Foundation. It can handle complex query syntax with modifiers to specify fuzzy searches, wildcards, quoted phrases. Also, a large number of morphological analyzers was created for Lucene.

Microsoft Azure full-text search engine [10] uses cognitive search and relies heavily on the capabilities of Lucene. It is possible to use the simple query language and the Lucene extended query language. Azure Cognitive Search supports a wide variety of morphology analyzers for Lucene. Artificial intelligence in Azure is used to enrich indexing. It uses image and natural language processing. Natural language processing includes entity recognition, language recognition, key phrase extraction, and sentiment determination [10].

An important aspect of information retrieval research is ranking evaluation. The paper [11] describes the problem of combining the results of semantic search and keyword search for correct ranking using TF-IDF for keywords. Discounted Cumulative Gain (DCG) is often used to assess ranking [12, 13]. But it is necessary to obtain relevance values for documents from some source to use DCG. The purpose of this work is a comparison of capabilities of the semantic search system proposed by the authors and the Lucene library for various groups of queries, including modifiers. It is also necessary to evaluate the search results ranking of these two systems.

2. Semantic search method

A new system was developed using Java language to study the possibilities of semantic search (Figure 1), which includes semantic search (search package), a metagraph knowledge base management system (knowledge package) for knowledge representation and natural language processing pipeline (analyzer package). The evaluation package has been created for comparison of the Lucene information retrieval library and the implemented semantic search engine. StandardAnalyzer class from lucene-analyzers-common and RussianLuceneMorphology class from RussianMorphology library are used for morphological analysis in Lucene. The last class also implements morphological analysis in the natural language processing pipeline for semantic search.

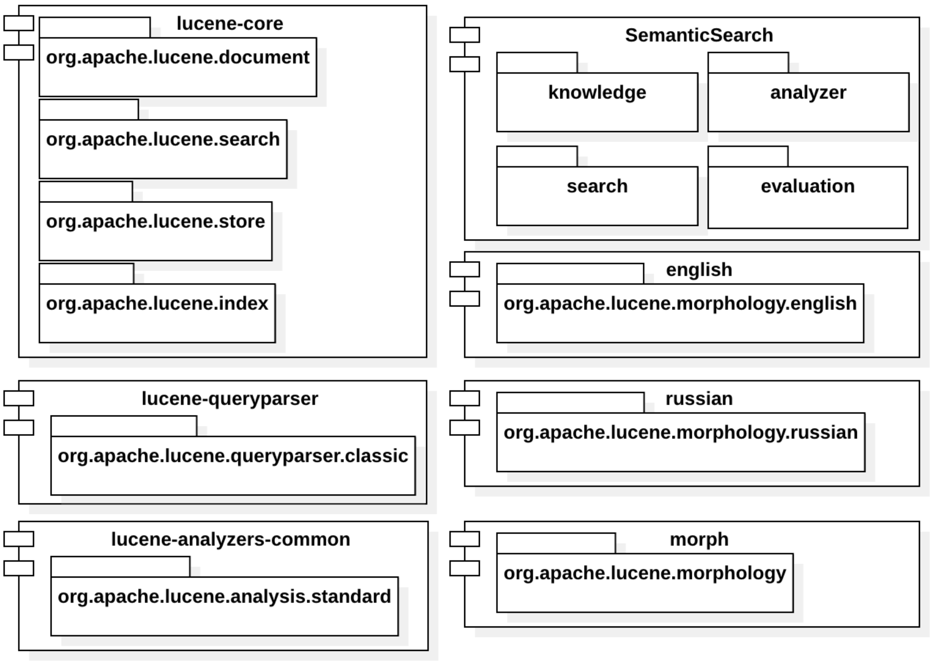
lucene-core
org.apache.lucene.document
org.apache.lucene.search
org.apache.lucene.store
org.apache.lucene.index
SemanticSearch
knowledge
analyzer
search
evaluation
english
org.apache.lucene.morphology.english
lucene-queryparser
org.apache.lucene.queryparser.classic
russian
org.apache.lucene.morphology.russian
lucene-analyzers-common
org.apache.lucene.analysis.standard
morph
org.apache.lucene.morphology

Figure 1. Component diagram.

It is proposed to use a semantic index to implement semantic search. Each concept corresponds to a set of documents which they mention. The search index based on concepts is presented in the following equation (1).

$$Index = \{< concept, \{i_{d_1}, .., i_{d_n}\} >\} \tag{1}$$

where the $concept$ is a concept associated with a set of document IDs $i_{d_1}, .., i_{d_n}$. To rank results using a new search index based on concepts we introduce modified TF and IDF measures in equation (2).

$$IDF'(concept) = \log \frac{N - n(concept) + 0.5}{n(concept) + 0.5} \tag{2}$$

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

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

3. Evaluating ranking results

Despite the fact that the precision of semantic search is higher than the precision of keyword search, in the developed system it does not affect the number of documents received. With complex concepts of several words more general concepts consisting of one word are considered. Therefore, the number of semantic search documents is the same as Lucene or it is less with limitations of natural language processing. But even for queries that have the same number of documents, the order of their results changes. Levenstein distance and the metric d_d proposed by the authors were used to compare the ranking results of the two search engines. The DCG metric is not used, since it requires values for the relevance of documents obtained from some source, and the considered Levenstein distance and metric d_d do not require them.

The characters used in this study are document identifiers, each representing one character. The analyzed data is a sequence of document IDs, ordered according to the calculated ranking score. The Levenstein distance and d_d were calculated for different types of queries for documents sequences of the semantic search system and the Lucene index. The value of Levenstein distance was normalized in each case to the total number of different documents in the results of the two systems.

The evaluation of documents ranking has some peculiarities. Documents cannot be repeated in the results of one system and documents can be located in different places in the list of results, including at positions far from each other. The Levenstein distance does not take into account the permutations of sequence elements; it is considered by the Damerau-Levenstein distance. But it only analyzes transposition of adjacent characters and it is more suitable for comparing strings in natural language. Therefore, to evaluate the ranking of the results of the two search systems, it was decided to use a different score, taking into account the proximity, which is equal to the sum of the scores for each unique element from the combination of both sequences in equation (3).

$$d_d = \sum_{c \in a \cup b} d_d(c) \tag{3}$$

$$d_d(c) = \begin{cases} 1, & if\ c \notin a\ or\ c \notin b \\ \dfrac{div}{length}, & if\ a_i = b_j = c \end{cases}$$

$$div = |i - j|$$

$$length = |a \cup b|$$

The score for one element is 1 if a character needs to be deleted or inserted, or div normalized to $length$ if the element is contained in both sequences, where div is the difference in the indexes of this element in two sequences, $length$ is the number of unique elements in two sequences. This metric will be the same as the Levenstein distance if the sequences do not contain common elements. In other cases, the Levenstein distance is less if the order of the elements in the sequences is the same, or more if this order is reversed.

4. Results

Open Corpora dataset was used as data for the study, which includes texts on various topics in several languages, including Russian and English. 4030 text files for indexing were obtained from a single xml file using the opencorpora-tools Python package. Testing was carried out on a computer with the following configuration: Intel Core i5, 6 cores, 3 GHz, 8 GB 2667 MHz DDR4, 256 GB SSD.

To compare the results of the semantic search engine and Lucene, a set of 130 queries in 11 groups was prepared: single word queries; short queries; short queries with prepositions; long queries; queries with verbs and adverbs; fuzzy queries; proximity queries; queries with Boolean operators; grouping queries with subqueries; queries with quoted phrases; wildcard queries. The first five groups of queries allow to evaluate the work of word search without special characters. The remaining six groups are needed to evaluate specific queries using modifiers.

The standard Lucene analyzer and the RussianMorphology morphological analyzer differ from each other. Therefore, the semantic search was compared with two versions of Lucene using different analyzers. The study was conducted for texts that contain words only in Russian in order to avoid the influence of words without analysis in other languages during the study. To estimate the execution time 100 iterations were carried out for each request and the average value was found. For each request 18 values were calculated, which were then averaged for each group of requests. To evaluate the ranking results, the Levenstein distance and metric d_d were calculated. These results are shown in table 1 and table 2.

Table 1. Results for queries without modifiers.

T_{sem} (ms)	T_{luc} (ms)	T_{lucRM} (ms)	Q_{sem}	Q_{luc}	Q_{lucRM}	P_{LC-SM}	R_{LC-SM}	P_{LC-RM}
R_{LC-RM}	P_{RM-SM}	R_{RM-SM}	L_{LC-SM}	d_{LC-SM}	L_{LC-RM}	d_{LC-RM}	L_{RM-SM}	d_{RM-SM}
Single word queries								
0,02	0,19	0,10	21,00	12,15	47,60	0,22	0,48	0,28
0,75	0,65	0,52	0,60	0,64	0,63	0,64	0,53	0,39
Short queries								
0,03	0,32	0,16	80,50	66,85	192,85	0,42	0,56	0,33
1,00	1,00	0,50	0,76	0,81	0,81	0,77	0,83	0,60
Short queries with prepositions								
0,03	4,89	0,57	118,70	1395,80	1410,20	0,91	0,08	0,99
1,00	1,00	0,09	0,98	0,94	0,68	0,09	0,98	0,92
Long queries								
0,07	5,38	0,70	179,60	1531,50	1556,30	0,96	0,11	0,98
1,00	1,00	0,12	0,98	0,91	0,79	0,13	0,99	0,89
Queries with verbs and adverbs								
0,00	0,24	0,11	0,00	53,20	166,30	0,00	0,00	0,26
0,70	0,00	0,00	0,70	0,70	0,70	0,68	0,90	0,90

Table 1 and table 2 use the following notation: T_{sem} - semantic search time, T_{luc} - Lucene StandardAnalyzer search time (hereinafter simply Lucene), T_{lucRM} - Lucene search time with RussianMorphology, Q_{sem} - the number of semantic search results, Q_{luc} - the number of Lucene results, Q_{lucRM} - the number of results Lucene RussianMorphology, P_{LC-SM} - semantic search precision compared to Lucene, R_{LC-SM} - semantic search recall compared to Lucene, P_{LC-RM} - Lucene RussianMorphology precision compared to Lucene, R_{LC-RM} - recall of Lucene RussianMorphology compared to Lucene, P_{RM-SM} - precision of semantic search compared to Lucene RussianMorphology, R_{RM-SM} - recall of semantic search compared to Lucene RussianMorphology, L_{LC-SM} - Levenstein distance for Lucene and semantic search, d_{LC-SM} - metric d_d for Lucene and semantic search, L_{LC-RM} - Levenstein distance for Lucene and Lucene RussianMorphology, d_{LC-RM} - d_d for Lucene and Lucene metric RussianMorphology, L_{RM-SM} - Levenstein distance for Lucene RussianMorphology and semantic search, d_{RM-SM} - d_d for Lucene RussianMorphology and semantic search metrics.

Table 2. Results for queries with modifiers.

T_{sem} (ms)	T_{luc} (ms)	T_{lucRM} (ms)	Q_{sem}	Q_{luc}	Q_{lucRM}	P_{LC-SM}	R_{LC-SM}	P_{LC-RM}
R_{LC-RM}	P_{RM-SM}	R_{RM-SM}	L_{LC-SM}	d_{LC-SM}	L_{LC-RM}	d_{LC-RM}	L_{RM-SM}	d_{RM-SM}
Quoted phrases								
0,06	0,04	0,06	62,40	0,00	0,30	0,00	0,00	0,00
0,00	0,10	0,30	1,00	1,00	0,30	0,30	0,90	0,90
Grouping queries								
0,00	0,20	0,14	0,00	30,20	88,50	0,00	0,00	0,35
1,00	0,00	0,00	1,00	1,00	0,80	0,75	1,00	1,00
Boolean operators								
0,01	0,14	0,10	33,20	21,40	65,90	0,15	0,31	0,36
1,00	0,40	0,29	0,88	0,89	0,74	0,75	0,87	0,76
Wildcard queries								
0,00	0,49	0,14	12,90	81,40	73,40	0,18	0,13	0,13
0,19	0,20	0,02	0,87	0,88	0,81	0,79	0,30	0,29
Fuzzy queries								
0,00	3,07	2,72	6,60	31,20	44,90	0,02	0,03	0,55
0,53	0,10	0,06	0,67	0,69	0,41	0,36	0,68	0,66
Proximity Search								
0,02	0,06	0,09	29,60	0,20	0,60	0,00	0,10	0,10
0,10	0,01	0,15	0,80	0,80	0,10	0,10	0,79	0,80

5. Discussion

The comparison between single word and short queries was influenced by the difference in the analyzers. A comparison of the columns for the number of results found for Lucene and Lucene RussianMorphology shows that for the second option this number is higher, although the standard analyzer returned a larger number of documents for wildcard queries. A more detailed analysis of wildcard queries shows that RussianMorphology does not return any results if the query contains only words with wildcard. Therefore, it is incorrect to use it for this group of queries.

For the groups of short queries with prepositions and long queries it can be seen that Lucene and Lucene RussianMorphology have almost the same number of results, but the ranking estimates using the Levenstein distance and the d_d metric are very different. The d_d value is much smaller, which corresponds to the close results of the two Lucene variants. This shows that it is more correct to use the d_d metric to assess the ranking.

Semantic search results are analyzed in comparison with Lucene RussianMorphology. The differences between single word queries and short queries are explained by the fact that natural language processing in semantic search cannot unambiguously analyze word forms corresponding to several parts of speech or lemmas. Short queries with prepositions as well as long queries contain frequently used prepositions and conjunctions, therefore, Lucene results are characterized by a large number of documents. The large number of resulting documents explains the long query execution times for Lucene in the case of queries with prepositions. The accuracy and recall of queries with verbs and adverbs are equal to zero, because the developed system does not analyze parts of speech other than nouns and adjectives.

It is proposed to solve these problems by indexing all unparsed words in addition to the semantic index. Prepositions and conjunctions can be excluded from the search, as they are very often used and change the ranking of documents from meaningful words. Another variant is to add new parsing rules to handle new parts of speech and word forms.

Other results were obtained for specialized query types containing Lucene query service characters, such as quotation marks, tilde, etc. Quoted phrases must be mentioned in the text as they appear in the query, and the semantic search index doesn't analyze and doesn't store them, therefore it finds erroneous results. Fuzzy search and wildcard search allow to find more documents when using Lucene, because it parses the relevant special characters and doesn't look for exact word matches. Proximity search requires analyze words in the text with a given distance from each other, which is not analyzed in semantic search. For these query groups, the semantic index cannot be used, it is necessary to parse the query syntax and use the Lucene index in this case.

Grouping queries use parentheses, and Boolean operators include special characters "+" and "-", corresponding to logical conjunction and negation operations. Since these characters are not processed in semantic search, for these groups it is necessary to parse the user query and build a tree for it before using the index.

6. Conclusion

The research carried out shows the issues of semantic search. The lack of complete natural language processing leads to limitations that can be overcome by increasing the number of parsing rules, as well as indexing words that are not parsed by natural language processing pipeline. To process queries containing special characters, it is necessary to parse the user queries, as well as the use of an additional index for phrases with quotes, fuzzy and other types of queries with modifiers. It was also shown that the proposed metric for ranking estimation shows better results in comparison with the Levenstein distance.

References

[1] https://blog.google/products/search/search-language-understanding-bert/ (date of access 20.09.2020)

[2] Sussna M 1993 Word sense disambiguation for free-text indexing Using a Massive Semantic Network *Proc. of the 2nd Int. Conf. on Information and knowledge management* pp 67-74

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

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

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

[6] Guha R, McCool R and Miller E 2003 Semantic Search *Proc. of the 12th Int. Conf. on World*

Wide Web

[7] Chernenkiy V, Gapanyuk Y, Nardid A, Skvortsova M, Gushcha A, Fedorenko Y and Picking R 2017 Using the metagraph approach for addressing RDF knowledge representation limitations *Proc. of the 7th Int. Conf. Internet Technologies and Applications 2017 Wrexham* pp 47-52

[8] Kanev A, Cunningham S and Terekhov V 2017 Application of Formal Grammar in Text Mining and Construction of an Ontology *Proc. of the 7th Int. Conf. Internet Technologies and Applications 2017 Wrexham*

[9] Kyriakakis A, Koumakis L, Kanterakis A, Iatraki G, Tsiknakis M and Potamias G 2019 Enabling Ontology-based Search: A Case study in the Bioinformatics Domain *19th Int. Conf. on Bioinformatics and Bioengineering 2019*

[10] https://docs.microsoft.com/en-us/azure/search/search-lucene-query-architecture (date of access 20.09.2020)

[11] Cohen S, Mamou J, Kanza Y and Sagiv Y 2003 XSEarch: A Semantic Search Engine for XML *Proc. of Conf. Very Large Data Bases 2003* pp 45-56

[12] Indumathi D, Chitra A 2011 A collaborative search with query expansion and result re-ranking *World Congress on Information and Communication Technologies 2011*

[13] Latifi S, Nematbakhsh M 2014 Query-independent learning to rank RDF entity results of SPARQL queries *4th Int. Conf. on Computer and Knowledge Engineering 2014*