Improving the Quality of Information Retrieval Using Syntactic Analysis of Search Query

Nadezhda Yarushkina[1][0000-0002-5718-8732], Aleksey Filippov[1][0000-0003-0008-5035], and Maria Grigoricheva[1][0000-0001-7492-5178]

[1] Ulyanovsk State Technical University, Ulyanovsk, Sev. Venec str., 32, 432027, Russia
jng@ulstu.ru, al.filippov@ulstu.ru, gms4295@mail.ru

Abstract. The paper describes the methods for linguistic analysis of search queries to improve the quality of information retrieval. The description of the parse tree in the form of a structure containing information about two or more related words with the indication of their parts of speech and location in the original request is used to the translation of search query in Elasticsearch Query DSL. Elasticsearch Query DSL has several disadvantages: the user may not know the features of Elasticsearch Query DSL, words joined by the OR operator is using by default in information retrieval. The using of the OR operator unnecessarily increases the recall and reduces the precision of information retrieval. Taking into account the features of Elasticsearch Query DSL and the information needs of the user allow to improve the quality of information retrieval.

Keywords: Information Retrieval, Syntactic Analysis, Search Queries.

1 Introduction

Information retrieval is the process of searching in an extensive collection of data some semi-structured (unstructured) material (document) that satisfies the information needs of the user.

Semi-structured data is data that does not have a clear, semantically noticeable and easily distinguishable structure. Semi-structured data is the opposite of structured data. The canonical example of structured data is relational databases. Relational databases are typically used by enterprises to store product registers, employee personal data, etc [1,2,3].

The quality of search in information retrieval systems is usually characterized by two criteria: recall and precision. The total count of found documents determines the recall. The ratio between the determines the precision found relevant documents and the total count of documents [2,3].

The quality of the search is affected by both the characteristics of the information retrieval system itself and the quality of the search query. An ideal search query can be formed by a user who knows well the domain area. Also, to form an ideal query, the user needs to know the features of current information retrieval system and their information retrieval query language. Otherwise, the search result will have the low precision or low recall values [1,2,3].

2 Main problem

To information retrieval, the user formulates a search query. The search query is a formalized way of expressing the information needs of information retrieval system users. Information retrieval query language is used for the expression of information needs. The syntax of information retrieval query language varies from system to system. Modern information retrieval systems allow entering a query in natural language in addition to an information retrieval query language [1]. Information retrieval system finds documents containing the specified keywords or words that are in any way related to the keywords based on the user search query. The result of the information retrieval system is a list of documents, sorted by relevance [2,3].

In this paper will consider the work of the proposed method on the example of the information retrieval system Elasticsearch [4].

Elasticsearch provides a full Query DSL [5]. The query string is parsed into a series of terms and operators. A term can be a single word – *quick* or *brown* – or a phrase, surrounded by double quotes –*“quick brown”* – which searches for all the words in the phrase, in the same order.

Operators allow to customize the search:

1. By default, all terms are optional, as long as one term matches. A search for *foo bar baz* will find any document that contains one or more of *foo* or *bar* or *baz*.
   The preferred operators are + (this term must be present) and - (this term must not be present). All other terms are optional. For example, this query:
   quick brown +fox –news
   states that:

   - *fox* must be present;
   - *news* must not be present;
   - *quick* and *brown* are optional – their presence increases the relevance.

2. Multiple terms or clauses can be grouped together with parentheses, to form sub-queries:
   (quick OR brown) AND fox.

Therefore, the Elasticsearch search algorithm has several disadvantages:

1. The user may not know the features of Elasticsearch Query DSL.
2. Words joined by the OR operator are used by default in information retrieval. The using of the OR operator unnecessarily increases the recall of information retrieval and reduces its precision.

It is necessary to develop a method of linguistic analysis and translation of a search query into a search query in a format of Elasticsearch Query DSL. The new format of search query allows to take into account the features of Elasticsearch Query DSL and improve the quality indicators (precision and recall) of information retrieval.

3 The method of Linguistic Analysis of Search Query for Improving Quality of Information Retrieval

The primary goal of the developed method of linguistic analysis and translation of a search query into a search query in a format of Elasticsearch Query DSL is the improvement of information retrieval quality. The main task is to select in the search query the groups of terms, united by some semantics.

3.1 The method of linguistic analysis and translation of a search query

The scheme of linguistic analysis of texts does not depend on the natural language itself. Regardless of the language of the source text, its analysis goes through the same stages [6,7]:

1. Splitting the text into separate sentences.
2. Splitting the text into separate words.
3. Morphological analysis.
4. Syntactic analysis.
5. Semantic analysis.

The first two stages are the same for most natural languages. Language-specific differences usually appear in the processing of word abbreviations, and in the processing of punctuation marks to determine the end of a sentence.

The results of the syntactic analysis are used to select in the search query the groups of terms, united by some semantics. To identify a noun phrase from a query identification of important query terms is necessary. It is also necessary to define the relationship between the terms of the query. A noun phrase or nominal phrase is a phrase that has a noun (or indefinite proper noun) as its head or performs the same grammatical function as such a phrase. Noun phrases are ubiquitous cross-linguistically, and they may be the most frequently occurring phrase type.

SyntaxNet as an implementation of the syntactic analysis process is used. SyntaxNet is a TensorFlow-based syntax definition framework that uses a neural network. Currently, 40 languages including Russian are supported. The source code of the already-trained Parsey McParseface neural network model that is suitable for parsing text is published For TensorFlow. The main task of SyntaxNet is to make computer systems able to read and understand human language. The precision of the model trained in the SinTagRus case is estimated at 87.44% for the LAS metric (Label Attachment Score), 91.68% for the UAS metric (Unlabeled Attachment Score) and determines the part of speech and the grammatical characteristics of words with an accuracy of 98.27% [8,9,10,11,12,13].

It is necessary to parse the search query to obtain a parse tree on the first step of the algorithm. To obtain data about the search query structure, dependencies between words and the types of these dependencies the resulting parse tree will be used.

The parse tree can be represented as the following set:

$$T = \{t_1, t_2, \ldots, t_k\}, \tag{1}$$

where k is a count of nodes in the parse tree;
t_i is a node of the parse tree, can be described as:

$$t_i = (i, w_i, m_j, c), i = \overline{1, k},$$

where i is an index of the word in search query;
$w_i \in W, W\{w_1, w_2, \ldots, w_k\}, W$ is a set of words of search query;
$m_j \in M, M = \{Noun,\ Propernoun,\ Verb,\ Adverb,\ Adjective,\ Conjunction,$
$Preposition,\ Interjection\}$ is a set of parts of speech for natural language;
c is an index of the word in the search query, that depends to the i-th word.

Thus, the search query is converted into a parse tree on the first step of the algorithm. For each word in the search query the part of speech, index of this word in the search query, and relations with other words of the search query are set.

The description of the parse tree in the form of a structure containing information about two or more related words with the indication of their parts of speech and location in the original request is used on the second step of the algorithm.

In the process of analysis of the input parse tree, the nodes that reflect the semantics of this query are selected. Search and translation of selected nodes into Elasticsearch Query DSL is executed using a set of rules. The translation process uses a set of rules. Rules are used to add special characters from the Elasticsearch Query DSL to the words of the search query. Also, stop words are deleted from search query during the translation. The result of the algorithm is a new search query that takes into account the semantics of information needs and features of the Elasticsearch Query DSL.

This algorithm can be represented as the following equation:

$$F^{Query} : (T, R) \rightarrow Q^*,$$

The input parameters of the function F^{Query} are the parse tree of search query T (eq. 1) and the set of rules R, and the result is a translated query Q^*.

$R = \{R_1, R_2, \ldots, R_n\}$ is the set of rules for searching elements in parse tree and their translation in Elasticsearch Query DSL.

Each rule can be represented as the following expression:

$$R_i(p, t_1, t_2, \ldots, t_m) = Q_j^*,$$

where p is a rule priority. The priority of the rule determines the order in which the rules are applied to the search query. Thus, the priority allows applying the more “complex” rules to define complex phrases first. If the more “complex” rule did not work, the rule with a lower priority is applied;
t_k is k-th element of the rule that allows to selecting the node (nodes) of the parse tree to process;
m is a count of elements in the rule;

$Q_j^* \in Q^*$, $j = \overline{1, q}$ is an element of translated query. Each element of a translated query contains the word or words of the original search query escaped by a symbol from the set of Elasticsearch Query DSL operators.

3.2 Examples of rules for linguistic analysis and translation of a search query

The formal description of the rule to search the noun phrase in the search query can be represented as follows:

$$R_{noun\_phrase} = \left(4, \langle i, w_i, Noun, c\rangle, \langle i+1, w_{i+1}, Adjective, i\rangle, \langle i+2, w_{i+2}, Adjective, i\rangle, \ldots, \langle i+d, w_{i+2}, Adjective, i\rangle\right) = \\ = +''w_{i+1}\ w_{i+2} \ldots w_{i+d}\ w_i'',$$

where d is a count of adjectives in the noun phrase.

Extraction of noun phrase from the parse tree of the search query finds one or more adjective that subordinates to the current noun tree node. The result of this rule is a noun phrase, escaped with ‘ +” ‘ at the beginning and ‘ “ ‘ at the end.

Each rule consists of the two sides: the left side and the right side. The left side is a template of a parse tree fragment. The right side is a string template. The method extracts fragments of the parse tree, matched by the pattern in the left side of the rule, and then converts them to the strings escaped by the symbols of Elasticsearch Query DSL, filling the string template.

The formal description of the rule to search the related nouns in the search query can be represented as follows:

$$R_{noun\_noun} = \left(3, \langle i, w_i, Noun, c\rangle, \langle i+1, w_{i+1}, Noun, i\rangle, \langle i+2, w_{i+2}, Noun, i\rangle, \ldots, \langle i+d, w_{i+2}, Noun, i\rangle\right) = \\ = +''w_{i+1}\ w_{i+2} \ldots w_{i+d}\ w_i'',$$

where d is a count of nouns that related to i-th noun.

Extraction of related nouns from the parse tree of the search query finds one or more noun that subordinates to the current noun tree node. The result of this rule is a set of nouns, escaped with ‘ +” ‘ at the beginning and ‘ “ ‘ at the end.

The formal description of the rule to search the proper noun that subordinates to the noun in the search query can be represented as follows:

$$R_{noun\_proper\_noun} = \left(2, \langle i, w_i, Noun, c\rangle, \langle i+1, w_{i+1}, Proper\ noun, i\rangle, \langle i+2, w_{i+2}, Proper\ noun, i\rangle, \ldots, \langle i+d, w_{i+2}, Proper\ noun, i\rangle\right) = \\ = +''w_i\ w_{i+1}\ w_{i+2} \ldots w_{i+d}',$$

where d is a count of proper nouns that related to i-th noun.

Extraction of proper nouns that subordinates to the noun from the parse tree of the search query find one or more proper noun that subordinates to the current noun tree node. The result of this rule is a set of nouns, escaped with ‘ +” ‘ at the beginning and ‘ “ ’ at the end.

The formal description of the rule to search the proper noun in the search query can be represented as follows:

$$R_{proper\_noun} = \left(1, \langle i, w_i, Proper\ noun, c\rangle, \langle i+1, w_{i+1}, Proper\ noun, i\rangle,\right.$$
$$\left.\langle i+2, w_{i+2}, Proper\ noun, i\rangle, \ldots, \langle i+d, w_{i+2}, Proper\ noun, i\rangle\right) =$$
$$= +'' w_{i+1}\ w_{i+2} \ldots w_{i+d}\ w_i\ ',$$

where d is a count of proper nouns that related to i -th proper noun.

Extraction of proper nouns from the parse tree of the search query finds one or more proper noun that subordinates to the current proper noun tree node. The result of this rule is a set of nouns, escaped with ‘ +” ‘ at the beginning and ‘ “ ‘ at the end.

The formal description of the rule to search the single noun in the search query can be represented as follows:

$$R_{single\_noun} = \left(1, \langle i, w_i, Noun, c\rangle\right) = +w_i.$$

As the main problem of the search subsystem of the SOM, users called the low quality of the information retrieval. This rule has a lower priority and is executed after the rules with a higher priority have been skipped only. The rule allows finding a node of parse tree with a part of speech noun that is not associated with other nodes with a part of speech noun, adjective, or proper noun.

The formal description of the rule to search the verb in the search query can be represented as follows:

$$R_{verb} = \left(1, \langle i, w_i, Verb, c\rangle\right) = +w_i.$$

This rule has the highest priority and does not overlap with any other rule. This rule allows finding a node of the parse tree with a part of speech verb.

4 Experiments

To test the method of linguistic analysis of search query proposed in this study some experiments were conducted. Figure 1 shows the primary form of application for translation of search query in Elasticsearch Query DSL.

The control “Original Query” is used to input original search query. The parse tree for original search query will be shown in table “Search query formatting elements” after pressing the button “Prepare query”. Each line of this table contains the name of the triggered rule and potential semantic group. Panel “Translation rules” allows the user to select necessary rules for translation the original search query to query in a format of Elasticsearch DSL. The resulting search query will be shown in control “Search query in new format” after pressing the button “Format”.

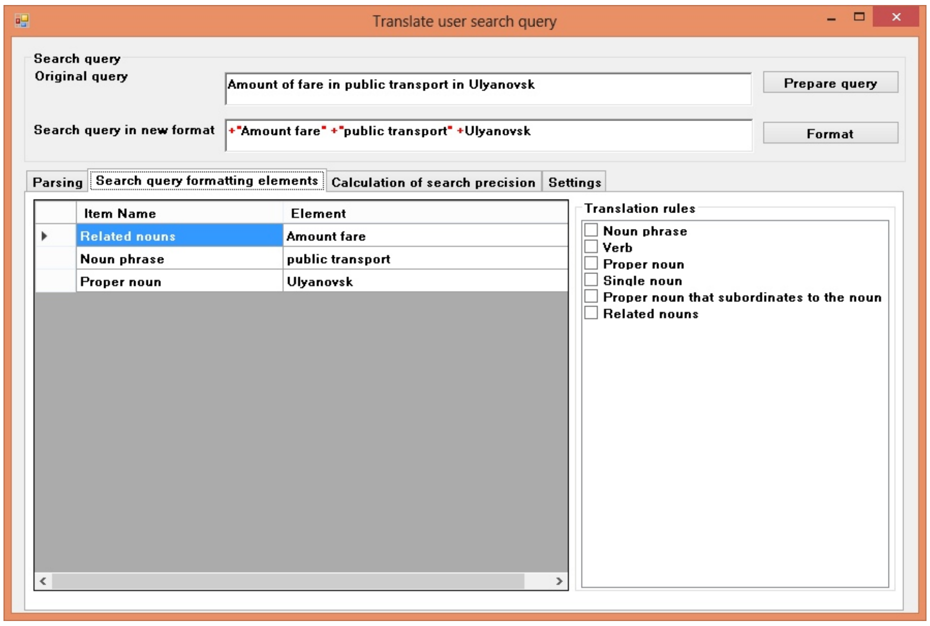
In this paper will consider the work of the proposed method on the example of the existing information retrieval subsystem of the system for opinion mining in social media (SOM). The SOM consists of the following subsystems:

1. Subsystem for importing data from external sources. This subsystem works mass media sites. Mass media loader retrieves data from HTML pages of mass media sites based on rules. The creation of own rules for each mass media is needed. The rule should contain a set of CSS-selectors. The ontology loader allows loading ontologies in OWL or RDF format into the data storage subsystem. Ontology is used for a description of the features of the problem area.

2. The data storage subsystem provides the representation of information extracted from mass media in a unified structure that is convenient for further processing. The data is stored in the context data sources, versions, etc. As database management systems are used:

   - Elasticsearch for indexing and retrieving data;
   - MongoDB for storing data in JSON format;
   - Neo4j for storing graphs of social interaction (social graph) and ontology.

3. The data converter converts the data imported from mass media into an internal SOM unified structures.
4. The OWL/RDF-ontology translator translates ontology into the graph representation [14].
5. The semantic analysis subsystem performs preprocessing of text resources. Also, this subsystem performs statistical and linguistic analysis of text resources.
6. The information retrieval subsystem finds objects related to a specific search query. In this case, the search query can be semantically extended using an ontology.



Translate user search query

Search query
Original query: Amount of fare in public transport in Ulyanovsk
Prepare query

Search query in new format: +"Amount fare" +"public transport" +Ulyanovsk
Format

Parsing | Search query formatting elements | Calculation of search precision | Settings

Item Name	Element
Related nouns	Amount fare
Noun phrase	public transport
Proper noun	Ulyanovsk

Translation rules
- ☐ Noun phrase
- ☐ Verb
- ☐ Proper noun
- ☐ Single noun
- ☐ Proper noun that subordinates to the noun
- ☐ Related nouns

Fig. 1. The main form of application for translation of search query in Elasticsearch Query DSL.

Posts and comments downloaded from mass media sites *ulpressa.ru*, *ulgrad.ru* and *mosaica.ru/ru/ul* are used as the dataset. The collection of documents is in Russian. The proposed rules are designed for the Russian language.

As the main problem of the search subsystem of SOM, users called the low quality of information retrieval.

Thus, the precision indicator of information retrieval is used to assess the quality of the proposed method. The recall and F-measure values are not used because the data storage subsystem of SOM contains the large count of documents.

Figure 2 shows the parse tree for search query: *“Стоимость проезда на общественном транспорте в Ульяновске”*. For example, will use the query *”Amount of fare in public transport in Ulyanovsk”* in English, which is close in meaning and structure to the query in Russian. The nodes of the parse tree are the words of the search query. Each node is assigned a part of speech.

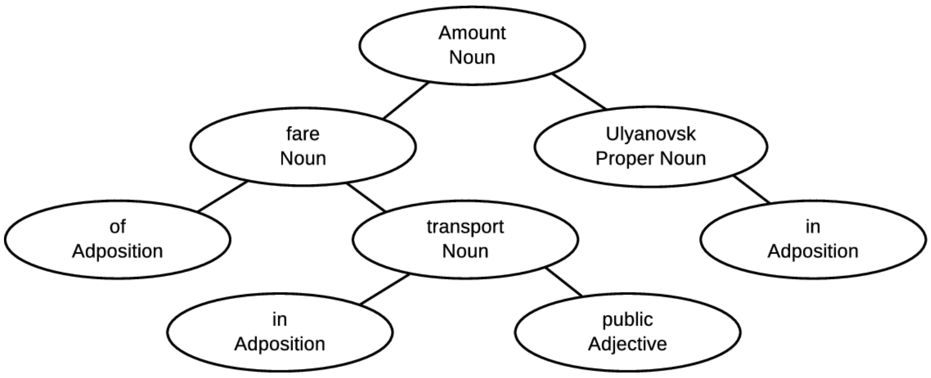


Fig. 2. The parse tree for the search query “Amount of fare in public transport in Ulyanovsk”.

After work of the algorithm, the significant elements were found in the parse tree (fig. 3). In the resulting tree (fig. 3), the nodes labeled as a rule with that they were found.

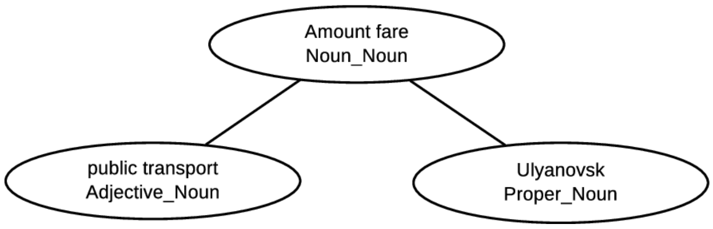


Fig. 3. The result tree for the search query *“Amount of fare in public transport in Ulyanovsk”*.

Thus, after linguistic analysis and translation the resulting search query for search query *“Amount of fare in public transport in Ulyanovsk”* is *‘+”Amount fare” +”public transport” +Ulyanovsk’*.

The precision value is calculated using the following expression:

$$P = \frac{a}{b}, \tag{2}$$

where a is a count of relevant documents in the search result;
b is a total count of documents in the search result.

For search query Q^O *“Amount of fare in public transport in Ulyanovsk”* the count of relevant documents in the search result of the information retrieval is 8. The total count of documents is 44857. Thus, the precision $P(Q^O)$ of the information retrieval for search query *“Amount of fare in public transport in Ulyanovsk”* is (eq. 2):

$$P(Q^O) = \frac{8}{44857} = 0{,}00018.$$

For search query Q^T *‘+”Amount fare” +”public transport” +Ulyanovsk’* translated from search query *“Amount of fare in public transport in Ulyanovsk”* using the proposed method the count of relevant documents in the search result of the information retrieval is 8. The total count of documents is 8. Thus, the precision $P(Q^T)$ of the information retrieval for search query *‘+”Amount fare” +”public transport” +Ulyanovsk’* is (eq. 2):

$$P(Q^T) = \frac{8}{8} = 1.$$

Thus, the using of proposed method improve the precision of information retrieval because of reducing the count of documents in the search result.

5 Conclusion

Elasticsearch Query DSL has several disadvantages:

1. The user may not know the features of Elasticsearch Query DSL.
2. Words joined by the OR operator are used by default in information retrieval. The using of the OR operator unnecessarily increases the recall and reduces the precision of information retrieval.

A method of linguistic analysis and translation of search query in Elasticsearch Query DSL allows improving the precision of information retrieval.

The search query is converted into a parse tree on the first step of the algorithm. For each word in the search query the part of speech, index of this word in the search query, and relations with other words of the search query are set. The description of the parse tree in the form of a structure containing information about two or more related words with the indication of their parts of speech and location in the original request is used on the second step of the algorithm. The result of the algorithm is a new search query that takes into account the semantics of information needs and features of the Elasticsearch Query DSL.

According to the results of 20 computational experiments, we can conclude: the use of the proposed method allows to increase the precision of information retrieval by an average of 18 times. The proposed algorithm can be used with any information retrieval system because it preprocessed the original search query, but does not change the system parameters or logic of information retrieval. However, the method requires adaptation to the features of the information retrieval query language of the current information retrieval system.

6 Acknowledgments

The study was supported by:

- the Ministry of Education and Science of the Russian Federation in the framework of the project No. 2.1182.2017/4.6. Development of methods and means for automation of production and technologicalpreparation of aggregate-assembly aircraft production in the conditions of a multi-product production program;
- the Russian Foundation for Basic Research (Grants No. 18-47-730035 and 16-47-732054).

References

1. Voorhees, E. M.: Natural language processing and information retrieval. In: Information Extraction, pp. 32-48. Springer, Berlin, Heidelberg (1999)
2. Manning C., Raghavan P., Schütze H.: Introduction to Information Retrieval. Cambridge University Press. (2008)
3. Miwa M. The Past, Present and Future of Information Retrieval: Toward a User-Centered Orientation. http://www.archives.go.jp/english/news/pdf/151106miwa_en.pdf. Accessed 20 Oct 2018
4. Elasticsearch. https://www.elastic.co/. Accessed 20 Oct 2018
5. Elasticsearch Query DSL. https://www.elastic.co/guide/en/elasticsearch/reference/2.3/query-dsl-query-string-query.html. Accessed 20 Oct 2018
6. SRILM - The SRI Language Modeling Toolkit. http://www.speech.sri.com/projects/srilm. Accessed 20 Oct 2018
7. Manning C., Schutze H.: Foundations of Statistical Language processing. In: The MIT Press. (1999)
8. Sboev A.G., Gudovskikh D.V., Ivanov I., Moloshnikov I.A., Rybka R.B., Voronina I.: Research of a Deep Learning Neural Network Effectiveness for a Morphological Parser of Russian Language. http://www.dialog-21.ru/media/3944/sboevagetal.pdf, Accessed 20 Oct 2018 (2017)
9. Chang J., Seefried J., Taylor S., Brandner A. SyntaxNet: Google's Open-sourced Syntactic Parser. http://www.sfs.uni-tuebingen.de/~keberle/NLPTools/presentations/SyntaxNet/SyntaxNet.pdf. Accessed 20 Oct 2018
10. Petrov S. Announcing SyntaxNet: The World's Most Accurate Parser Goes Open Source. https://ai.googleblog.com/2016/05/announcing-syntaxnet-worlds-most.html. Accessed 20 Oct 2018
11. Weiss D., Petrov S. An Upgrade to SyntaxNet, New Models and a Parsing Competition. https://ai.googleblog.com/2017/03/an-upgrade-to-syntaxnet-new-models-and.html. Accessed 20 Oct 2018
12. Alberti C., Orr D., Petrov S. Meet Parsey's Cousins: Syntax for 40 languages, plus new SyntaxNet capabilities. https://ai.googleblog.com/2016/08/meet-parseys-cousins-syntax-for-40.html. Accessed 20 Oct 2018
13. Eberle K. Natural Language Tools - Test and Comparison. http://www.sfs.uni-tuebingen.de/~keberle/NLPTools/slides/slides01.pdf. Accessed 20 Oct 2018

14. Yarushkina N., Filippov A., Moshkin V.: Development of the Unified Technological Platform for Constructing the Domain Knowledge Base Through the Context Analysis. In: Creativity in Intelligent Technologies and Data Science, pp 62-72 (2017)