Chapter

Methods of Russian Patent Analysis

Dmitriy Korobkin, Sergey Vasiliev, Sergey Fomenkov and S.G. Kolesnikov

Abstract

The article presents a method for extracting predicate-argument constructions characterizing the composition of the structural elements of the inventions and the relationships between them. The extracted structures are converted into a domain ontology and used in prior art patent search and information support of automated invention. The analysis of existing natural language processing (NLP) tools in relation to the processing of Russian-language patents has been carried out. A new method for extracting structured data from patents has been proposed taking into account the specificity of the text of patents and is based on the shallow parsing and segmentation of sentences. The value of the F1 metric for a rigorous estimate of data extraction is 63% and for a lax estimate is 79%. The results obtained suggest that the proposed method is promising.

Keywords: patents, information extraction, SAO, ontology, prior art patent search

1. Introduction

From year to year, the number of patents and patent applications is increasing. The escalating applications flow, and more than 20 million world set of granted patents (from 1980 to 2015) increase the time that patent examiners have to spend to examine all incoming applications. Also, automation development of inventions has been gaining momentum, and computer-aided invention (CAI) systems are used to search for new technical solutions. The success of such systems largely depends on the completeness of the ontologies of the subject areas and the fullness of the various knowledge bases that allow generating new technical solutions.

The task of prior art patent search and information support for new technical solution synthesis can be seen as the task of extracting the subject-action-object (SAO) semantic structures. Patent claims are considered to be a direct source of data for retrieval. They express the essence of the invention, and therefore it is of the greatest interest for the SAO extraction.

In the paper [1], authors proposed a methodology to solve the problem of prior art patent search, consisting of a statistical and semantic analysis of patent documents, machine translation of patent application, and calculation of semantic similarity between application and patents on the base of subject-action-object (SAO) triples. On the step of the semantic analysis, the authors applied a new method for building a semantic representation of SAO on the base of meaning-text theory [2]. On the step of semantic similarity calculation, the authors compare the SAOs from the application and patent claims. The results of checking the Russian- and English-language version of the semantic analyzer showed a low recall value for Russian patents—this is due to the complexity of Russian grammar.

Current approaches to extracting structured data and natural language processing (NLP) tools are poorly oriented to work on an array of Russian patents. In this regard, it is necessary to develop new efficient methods for extracting data from Russian patents.

The task of prior art patent search and information support for new technical solution synthesis can be seen as the task of extracting the subject-action-object (SAO) semantic structures. In particular, the elements of the design of a technical object and the relationship between them are considered as the source of information support.

Claims are considered to be a direct source of data for retrieval. They express the essence of the invention and are based on a description; therefore, it is of the greatest interest for the extraction of technical information. In this case, the object of analysis is an array of Russian-language patents.

This paper analyzes the application of the current NLP tools and data extraction approaches in the context of the current task. The article offers a new method of extracting SAO structures and constructing a graph of structural elements as part of solving the problem of prior art patent search and information support for the synthesis of new technical solutions.

2. Research background

Various systems [3–6] are known for processing English-language patents, including the use of the SAO formalism to extract various concepts. In [6], a tree syntax analysis was applied with a separate identification of the subject, the action, and the object to improve the quality of parsing. The authors state the following averaged values of precision and recall: 0.8058 and 0.8446, respectively. At the same time, the project was implemented in the GATE system, which is poorly suitable for the Russian language. In [7], the SAO structures are extracted based on the rules using the Stanford parser software, but there are no ready-made models for the Russian language. Papers of [4, 8] process patents using linguistic markers (specific verbs and nouns) and lexical-syntactic patterns, while [8] notes the need to study the structure of patent documents to improve the quality of data extraction. The emphasis is mainly on rule-based systems, since for statistical analysis systems, a lot of markup data is obviously needed. This causes additional development difficulties.

The logic of constructing a claim of the invention of the European type, applied in the Russian Federation, includes statement of the formula, starting with the generic term, and reflecting the purpose of the invention; after the expression, “including,” “containing,” or “consisting of,” sets out the restrictive part of the patent claim, including the features of the invention, coinciding with the signs of the closest analogue and then after the phrase “characterized in that,” its distinctive part is introduced, including the features that distinguish the invention from the closest analogue [9]. The formula is written in one sentence.

The specific style of writing a patent claims complicates the work of existing parsers, highlighting the following problems:

- The length of the sentences of the claims is excessively long.

- The claims are described predominantly with noun phrases, often with a chain of definitions and participial constructions.

- The availability of specific words and constructions can affect the correct operation of NLP tools.

Considering the system for extracting structured data from Russian-language texts, the development of Yandex Tomita-parser [10] is most often distinguished. Extraction takes place according to the rules, written in an extended context-free grammar.

In [11], the authors emphasize the advantages of Tomita-parser in the tasks of extracting named entities; the declared value of the F1 metric is 78.13%. In [12], this tool is successfully used to extract metadata from full-text electronic Russian-language publications with correct extraction of 86.7% of metadata. In [13], the Tomita-parser is used to extract from the patents SAOs describing the so-called technical functions.

However, due to the specifics of writing the claims on the invention, there are difficulties in using the Tomita-parser:

- Homogeneous subjects are not extracted using coordination mark of subject and predicate [10], since coordination of the whole bundle is not taken into account.

- Writing grammars for all possible parse trees with regard to the free word order in the Russian language is a laborious procedure.

- Incorrect identification of actants (erroneous determination of the cases of words, the need to take into account the valencies of the verb).

Despite the expressive power of context-free grammars and the Tomita-parser tool, the question of organizing the effective extraction of SAO structures from the claims remains open.

Of the available systems of syntactic and semantic-syntactic analysis, for which there is an opportunity to work with the Russian language, we can distinguish Link Grammar Parser [14], MaltParser [15], and UFAL UDPipe [16].

Link Grammar Parser is an open-source grammar parser. In [17], a system is described for extracting technical functions in SAO format from English-language patents using context-dependent grammar on Link Grammar Parser markup. At the same time, the results of experiments are presented (with an extraction precision of up to 0.85 and recall up to 0.78). The quality of work of Link Grammar Parser with the Russian language is the open question.

MaltParser is a tool for working with dependency trees, allowing to train a model on the annotated corpus and build trees for new data based on it. It is possible to train an own model on the annotated corpus of the Russian-language SynTagRus, openly laid out at the Universal Dependencies project [18]. The prepared morphological markup in the format CoNLL-U [19] is supplied to the analyzer input, which is supplemented in the process of analysis by the arrangement of vertices and types of links. In [20], estimates of the quality of semantic analysis of tools based on MaltParser are given; the value of the F1 measure reaches 71.0%.

UDPipe is a trained tool for tokenization, tagging, lemmatization, and dependency parsing. The project is open and actively developing. Trained models are available for a variety of languages, including Russian. The format of the input and output data is presented in the CoNLL-U markup. For the model trained on the SynTagRus package, the markup accuracy (on raw text) for all tags, as well as for the labeled attachment score (LAS) and unlabeled attachment score (UAS) metrics, is 93.2, 85.3, and 87.9%, respectively.

One of the main obstacles to the use of these systems is the excessive length of the sentences of the claims and, in particular, the considerable distance of homogeneous members from the binding predicate. The latter has a significant impact on the completeness of the extraction of objects.

Omitting the issues of training models, data selection, analysis tools, morphologists, and other things, in general, it can be concluded that the syntax and semantic-syntactic markup tools (as reviewed) work extremely unstable on long texts. It is impossible to use direct analysis results for efficient data extraction. At the same time, it is this structure of the description that prevails in the texts of the patent claims.

A possible solution in reducing the complexity of parsing text is its segmentation. The work [21] presents an algorithm for the segmentation of the text of the claims, based on the specific markers of the division of sentences. This approach reduces the length of the analyzed segment of the proposal but is not sufficient for a full analysis due to the possible absence of signs of separation in complicated sentences.

One of the most extensive works on the full segmentation of sentences in Russian texts can be considered the works of T. Yu. Kobzareva. The book [22] describes the elements of grammar and algorithmic solutions of the system of shallow parsing. In general, the system is set up to analyze the proposals of the Russian language, but it is rather cumbersome.

In [23], on the basis of the analysis of the existing parsers, the author builds the analyzer version, the segmentation module of which, by its functionality, closely matches the task at hand.

There are various approaches to segmentation, for example, taking into account the valences of verbs, using patterns of bases, etc. However, these solutions are universal with respect to texts and therefore do not have specific rules for the analysis of the claims of the invention and in some cases are somewhat redundant.

When building own segmentation and data extraction modules (low-level approach), it is necessary to have reliable data on the morphology of the sentence: analysis of the form of incoming words and their grammatical categories (gender, case, etc.). Among the available Russian-language analyzers that can be identified are TreeTagger, MyStem, TnT, pymorphy2, and FreeLing. A number of works [24–26] are devoted to the comparison of morphological analyzers of the Russian language. In the works [25, 26] on the quality of determining the parts of speech (POS), TreeTagger is best distinguished, but they do not take into account the work of the Yandex MyStem tool. In [24], already taking into account MyStem, the latter was chosen as the best tool with the F1 metric value of 94%. At the same time, an experiment is being conducted to improve the quality of parsing by combining tools (correcting MyStem with pymorphy2 with an increase in F1 to 95%). However, the full use of the analyzer MyStem is difficult due to the issuance of grammatical signs in the form of an unordered list of alternatives. At the same time, there are no attributes for the correct choice of a single grammatical form; the alternatives simply follow in alphabetical order. The TreeTagger morphoanalyzer is devoid of such a feature—the grammar is issued without alternatives. The disadvantage of TreeTagger is that its speed is somewhat lower than the speed of processing MyStem on large texts.

Having considered certain aspects of the processing of Russian-language texts, we summarize the problems that arise when extracting structured data (SAO structures) from texts of Russian-language patents when solving the problem of prior art patent search and information support for automated invention:

- The claims are written in a specific sublanguage of patents that impede text analysis (specific terminology).

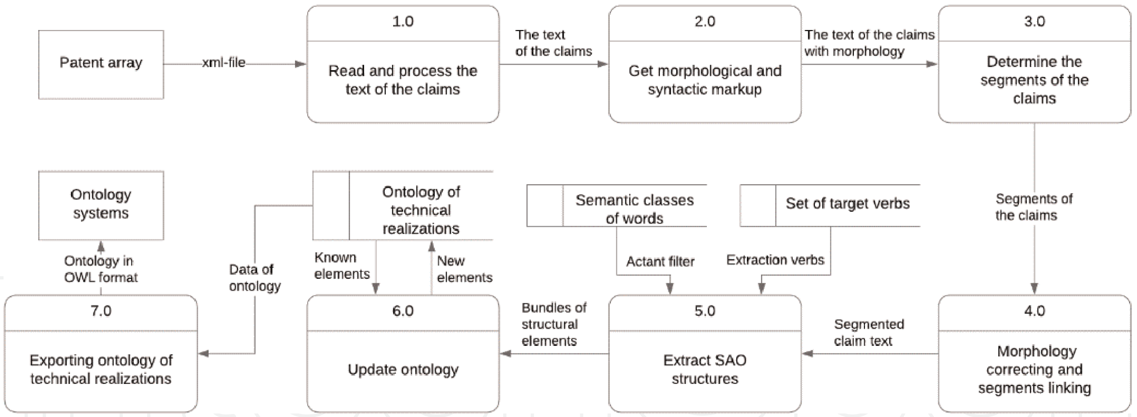

Patent array
xml-file
1.0
Read and process the text of the claims
The text of the claims
2.0
Get morphological and syntactic markup
The text of the claims with morphology
3.0
Determine the segments of the claims
Segments of the claims
4.0
Morphology correcting and segments linking
Segmented claim text
5.0
Extract SAO structures
Semantic classes of words
Actant filter
Set of target verbs
Extraction verbs
Bundles of structural elements
6.0
Update ontology
Ontology of technical realizations
Known elements
New elements
Data of ontology
7.0
Exporting ontology of technical realizations
Ontology in OWL format
Ontology systems

Figure 1.
DFD chart of the proposed system.

- The structure of the claims is poorly suited for parsing by ready-made syntactic analyzers (given the prevalence of sentences and the presence of multiple participle clauses).

- The structure of the claims is not fully decomposed by simple heuristics.

- The available NLP tools will certainly be mistaken in the analysis, so relying on one tool in the processing of complex structures is not always rational.

The presented limitations make it difficult to create an efficient data extraction system from patent claims using publicly available NLP tools (without considering complex commercial or closed systems).

However, given that the texts of patents are written in a template (not only in the structure of sections but also in the structure of phrases), it is possible to combine NLP tools for mutual correction of analysis errors; the authors suggest the following approach in processing the text of the claims, including the steps:

- Segmentation of the patent claims based on shallow parsing

- Organization of the extraction of SAO structures based on the valence bond theory

- Post-processing of SAO structures with linking concepts to a common repository

Next, we will consider building a system based on the findings.

The main stages of processing and data flows of the proposed system are presented in **Figure 1**.

In this research, the stages of the ontology organization are not considered (actions 6.0, 7.0). The end result will be the extraction of SAO structures and the construction of a graph of the elements of the invention.

3. Method description

The unit of extraction is the predicate-argument construction in the form of the SAO structure, which semantically describes the structural elements of the device and the connections between them.

The following instruments of NLP in the experimental mode are used:

- UDPipe (tokenization, tagging, lemmatization, and syntactic analysis)
- MyStem (lemmatization, correction of morphology)
- pymorphy2 [27] (declination of word forms)
- Chanker of noun phrases of Russian [28] (to define the boundaries of actants)

The ultimate goal of extracting data is to build a related graph of structural elements of the invention, reflecting the structure of the invention itself. Let us consider in more detail each of the stages.

3.1 Segmentation of patent claims

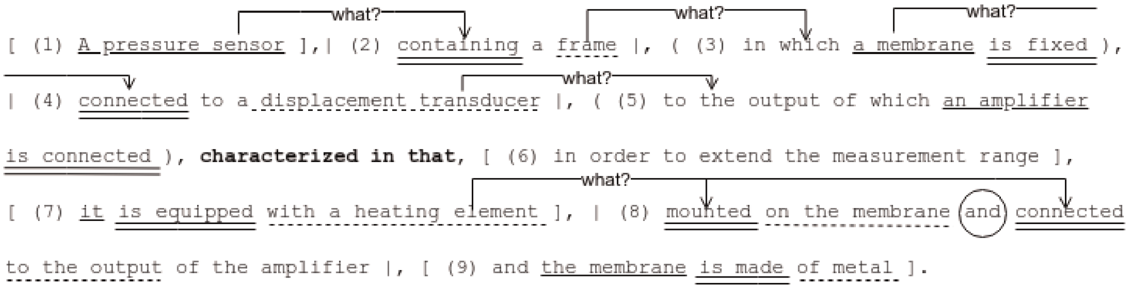
On the basis of a well-established patent language, the authors made the assumption that it is possible to identify certain patterns of segments in a patent claim and that shallow parsing with a number of heuristics will be enough to successfully extract the necessary structures.

The segment of the independent claim will be called the element of the sentence, which is indicated by punctuation (comma and/or semicolon). In the Russian language, such segments can act as simple main sentences and be subordinate clauses, participle and verbal adverb phrases, isolated definitions, etc. In addition, one should take into account the logical division of the claim into the restrictive and distinctive parts (if such separation exists), usually demarcated by the construction of "characterized in that."

The purpose of segmentation is to identify the linear structure of the proposal of the claims for the subsequent extraction of SAOs based on the analysis of the morphological and linear combinatorial characteristics of the text. Let us explain it with the example of the text of the patent "Pressure sensor," the syntactic analysis of which is presented in **Figure 2**.

In parentheses are marked linear numbers of segments without taking into account the demarcation design "characterized in that." At the same time, the verb forms without specifying the role in the sentence are highlighted with a double line.

The sentence includes generic term (1); participle clause (2, 4, 8); subordinate-attributive (3, 5); and simple two-part (7, 9). The tasks of segmentation include the identification of segments by composition and the definition of the relationship between the main and dependent words. In the future, this information is necessary for the direct extraction of SAOs.



[(1) A pressure sensor],| (2) containing a frame |, ((3) in which a membrane is fixed), | (4) connected to a displacement transducer |, ((5) to the output of which an amplifier is connected), **characterized in that**, [(6) in order to extend the measurement range], [(7) it is equipped with a heating element], | (8) mounted on the membrane and connected to the output of the amplifier |, [(9) and the membrane is made of metal].

Figure 2.
Parsing sentence.

Type of segment	Symbol	Example of a segment
Noun phrase	N	"contact electric current modulator"
Participle clause	V_N	"containing vibration transducer"
Independent segment	N_V_N	"it is equipped with an additional vibration converter"
Isolated attribute	A0	"identical to the main"
Rather independent segment	(N_) which_V_N	"which input is the input of the device"
Dash instead of a predicate	N_-_N	"and the cathode through resistance—with the anode"
Comparative turns	AV	"less"
Subordinate clauses	SC	"that they have sufficient fatigue strength"
Verbal adverb phrase	V<C>	"preventing the deformation of the first destructive elements"
Service segments	SEPARATOR	besides; and; and also; at the same time
	NAMED	"called decaying elements"
	PURPOSE	"for the purpose of range extension of measurements"

Table 1.
Types of segments of the test sample claims.

To identify the nature of writing segments and formalize the definition of their types, a set of 14 main claims was taken, mainly from different sections of the international patent classification (IPC). The total number of segments in the sample was 187 elements.

After a detailed manual analysis, the following conditional segment types were derived (see **Table 1**).

At the same time, the symbol for the segment was formed from acronyms, taking into account the basic structure of the segment, i.e., the POS and/or elements that characterize it: noun (N); verb (V); adjective (A); "which" is a demonstrative pronoun in subordinate clauses; adverb (AV); and others. The legends of the segments in the future act as markers in the system.

Apparently from the results of the analysis, all the segments met a finite number of fairly typical sentence constructions. Of course, it is not possible to thoroughly analyze the entire array of patents manually. Therefore, the authors made the assumption that the number of types of segments (in terms of the overall structure of writing) of a larger mass of the claims is also limited.

Since the segmentation is carried out in the context of the task of extracting predicate-argument structures, the elements of this extraction are partially embedded in the process in the form of highlighting nonsignificant (service) segments and the target semantic classes of verbs.

Semantic classes of verbs are needed to find to SAO target ligaments, potentially describing constructive elements and the connections between them. The samples of such verbs are presented in **Table 2**.

The overall sequence of operations for the segmentation of the claims is represented by the activity diagram in **Figure 3**.

1. Text preprocessing includes the following steps:

   - Deletion of introductory words: "for example," "at least," "however," etc.

   - Removal of brackets with all contents: for example, references to figures "(1), (12)"

Semantic class	Verbs
Existence of a structural element	contain (on); equip; include (between; in; with; in;); have; supply; enter (into); consist (from)
Inter-element couplings	set (on; as); connect (to); fasten (in; on); place (in; on)

Table 2.
Semantic classes of verbs.

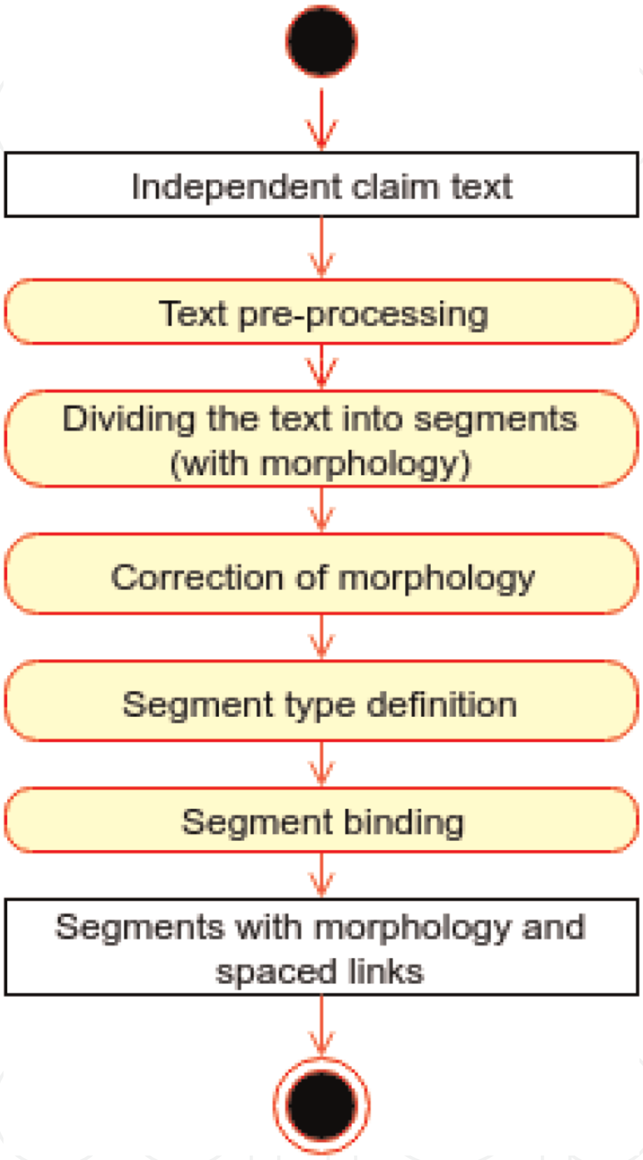


Figure 3.
The segmentation process of the claims.

- Deletion of the claim number at the beginning of the text ("1. Magnetic gear … ")
- Removal of html tags (<. *?>) in case of reading from xml documents
- Addition of spaces to slashes (to correct the morphology to "and/or")
- Removal of multiple spaces and duplicate commas

2. The process of obtaining initial segments is associated with sequential fragmentation of the text by punctuation marks with auxiliary treatments. A

specific feature of the stage is the final fragmentation based on the results of the syntactic analysis obtained after using the UDPipe tool. Tokenization is also performed by this tool. This decision is justified by the need to take into account the context for better analysis: submitting partially fragmented text to the analyzer input (based on strong signs of separation—e.g., different parts of the claims or a semicolon), but without carrying out fragmentation on commas, it is possible to receive more adequate syntactic marking; analysis of segments completely isolated from each other potentially also leads to errors.

The division of the text into segments includes the steps:

- Division of the text according to the pattern, "characterized in that," while the segment is marked with a conditional part number: (0, all segments belong to one part (claim without division); 1, segment of the restrictive part; 2, segment of the distinctive part)
- Division of the text by a semicolon
- Insertion of surrogate references to formulas and complex tokens (e.g., "S = W = 0.00042λ") to reduce segmentation errors and morphological analysis
- Obtaining the primary morphology from the UDPipe tool and segmentation by commas

3. The next step is the process of correcting morphology. If a segment token refers to parts of a verb, adjective, or adverb speech, then a part of speech is checked using MyStem. Moreover, if the result does not coincide with the initial one, then the morphological information of the token is updated taking into account the conversion of tags from MyStem into UDPipe format (CoNLL-U).

One of the fundamental stages of segmentation is the determination of the type of segment (see **Table 1**). The type of segment allows you to introduce certainty in the use of certain heuristic processing in the future. This problem is solved in two steps:

a. Determining a segment pattern

b. Determining the type of segment by pattern using a finite state machine (FSM)

The definition of a segment pattern is based on finding out its main composition, which is based on the priority of the presence of POS. The dictionary of symbols in the template is presented in **Table 3**.

A generalized algorithm for determining the pattern of a segment by the POS found in it is shown in **Figure 4**.

An example of the output templates and types for the segments is an example from **Figure 2**.

Segment (8): "mounted on a membrane and connected to the output of an amplifier" ("установленным на мембране и подключенным к выходу усилителя" in Russian).

Parts of speech (ru): [VERB ADP NOUN CCONJ VERB ADP NOUN NOUN]

Template: "V <P> _N_CC_V <P> _N"

Type: "V_N"

Symbol	Part of speech	Note
N	Noun	Index 0 means accusative or a nominative case
V	Verb	Possible litters: <P> participle; <F> finite verb; <C> adverbial participle; <I> infinitive; <U> undefined
A	Adjective	Index 0 means the top of the sentence (indirect attribute)
SC	Subordinate conjunction	—
CC	Coordinative conjunction	—
PR	Pronoun	Index 0 means the nominative case Direct replacement of the word “which” is possible
AV	Adverb or numeral	—
X	Formula	For the case of explanation of the formulas
—	Dash	Verb skip or explanation

Table 3.
Template dictionary.

The definition of a segment type by the template is implemented using an FSM. The principle is to search for elements of the segment template and search for specific sequences of POS corresponding to the types of segments.

4. Let us describe the process of binding of segments. The main binding mechanism in the Russian language is coordination according to gender, number, and case. Therefore, the segmentation is carried out on the data of the extracted morphology, and the task is reduced in finding this agreement between the words in the sentence. At the same time, segment types limit the set of binding rules.

   Proceeding from a linear logic of the description of the formula, it is possible to a connection of segments in one straight pass (from left to right). An exception is the case of a rupture of the sentence by an enclosed turnover. As a rule, the second part of a broken sentence begins with a finite verb. When finding such a segment, it is necessary to find the closest missing segment of the noun group to the left. The assembly of such segments precedes the main straight passage.

   Next, each segment type is processed. It is necessary to explain the mechanism of binding to the predicate forms of homogeneous members and constructions: <`(1) AC voltage regulator, containing (2) a discrete-action regulator, connected between the input and output terminals, (3) an interval converter into a code, whose information input is connected to the output terminal, a synchronizing input - to the input terminal, ..., as well as (4) a logic device`>.

   The generic term is “AC voltage regulator” (1). The participle “containing” combines the enumerated structural elements of the device (“regulator” (2), “converter” (3), and “logic device” (4)). However, some homogeneous members (in the example it is “logic device” (4)) may be located far away from the describing segment (participle “containing”).

To solve this problem, firstly, a dictionary of predicates is introduced (see **Table 2**), which is semantically pointing to the composition of the object (e.g.,

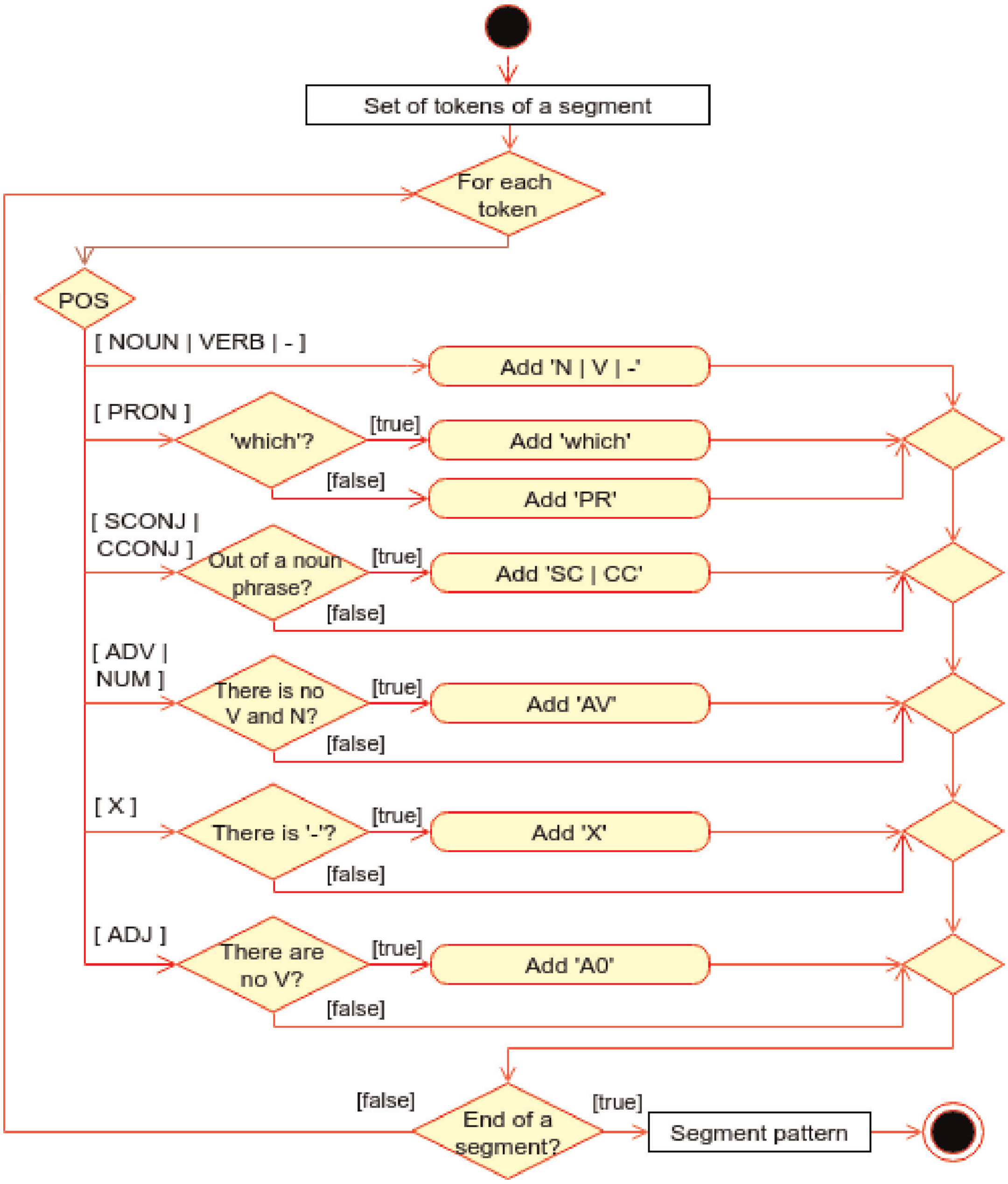


Figure 4.
Segment pattern generation sequence.

"contain," "include," "have," etc.); secondly, the mechanism of binding to predicates is introduced.

Anchor points (predicates) are tracked as structures:

1. Last compositional verb (LCV), searched by dictionary: LCV = (<segment_id>, <verb_id>).

2. Last active verb (LAV), remembers the last predicate in the corresponding types of segments, since homogeneous terms can follow: LAV = (<segment_id>, <verb_id>, <noun_case>). The LAV structure stores information on the case of a noun following the selected verb for anchoring correction.

Example of binding to predicates:

```
 containing[LCV] an amplifier (Nom/Acc case), connected[LAV] to
the source (Dat case) of the power supply, filter (Nom case)...
```

A brief description of each type of segment is presented in **Table 4**.

The result of segment linking for an example (**Figure 2**) is shown in **Figure 5**.

A pseudo-vertex (ROOT) is introduced on the graph, to which all independent, initial, or unrecognized segments are attached. According to the graph, it can be noted that the participial revolutions (2, 4, 8) are linked to the parent segments. Segments with the demonstrative pronoun "which" (3, 5) are also attached; the type of the connection "spec" is marked. In general, the entire convolution is performed correctly.

Segment type	Heuristic rules
N	The first segment of the entire claims is declared independent Otherwise, a sequential attempt to bind to LAV and LCV
V_N	In the presence of participle phrase, which starts with a co-coordinating conjunction, an attempt is made to bind to LAV and LCV. Check on the construction of V (plural) N (singular): in the presence of LCV, this segment will be homogeneous. Binding of the first predicate to the noun in the agreement of gender, number, and case. Rewriting LAV and LCV structures
N_V_N	The default is an independent segment. Check for a possible type determination error by the presence of V<P> at the beginning of the segment followed by linking to the main word by agreement
A0	In the case of the beginning of a segment with an adjective, a binding is made to the noun in the previous segments as agreed A segment can be declared independent if it starts with a noun and ends with the adjective A0
N_which_V_N, which_V_N	The specifying segment contacts a noun in the previous segment under the approval of a gender, number, and case of the word "which"
N_-_N	A segment with a missing predicate is bound to the previous segment if it is not a description of the formula
AV/SC	Adverb or comparison/complex sentence. The default binding to the previous segment
V<C>	The verbal adverb phrase is looking for a point of attachment in the form of a segment with a finite verb
PURPOSE	Segments describing the purpose of constructions depending on the form of the record are tied to the previous or next segment ("intended ... " or "for the purpose of ... ")
NAMED	The names of the various elements have a mechanism for linking the participial phrases ("called ... ")

Table 4.
Heuristic rules for segments.

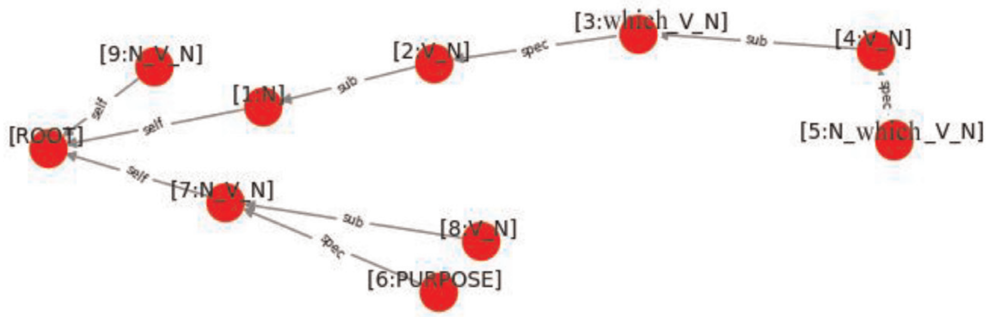


Figure 5.
The graph of segments for an example (**Figure 2**).

3.2 Extracting SAO primary structures

The extraction of predicate-argument constructions is based on the results of segmentation and valency of the target verbs (see **Table 2**). The valence of the verb determines the number of possible arguments. It is enough to define the "subject" and "object" of the structure extracted.

The subject can be determined by the following cases (see Example 1), by the nominative case (1) and by the word of the participle being defined (2), or it can be completely absent (3), implying a generic term.

Example 1. The definition of the subject in the segments:

1. The magnetic gearbox is equipped with a rotor …

2. … magnetic gearbox, equipped with a rotor …

3. … characterized in that …, a rotor is introduced, …

The definition of the object is somewhat complicated due to the lack of explicit universal expression in the predicate arguments. The role of an object in a predicate corresponds to a specific case, which can be formed using prepositions.

Consider the cases of determining the object from Example 2: accusative case without preposition (1), dative case formed with the preposition "to" (2), and several actants with prepositions (3); in the latter case, the preposition "through" generates an adverb (connected as?—"through a non-magnetic ring"), and the target actant is a combination with the preposition "with" (connected with what?—"with the body").

Example 2. The definition of the object in the structure:

1. including shaper

2. connected respectively *to* the output of the amplifier

3. connected *with* the body *through* a non-magnetic ring

It should also be taken into account that the predicate can be expressed by a verb with a compound preposition (e.g., "made in the form of"). In addition, information on binding to the predicate can already be partially determined at the end of the segmentation.

To account for the valences of the target verbs, the following structure is introduced:

```
verb (obj_valence, add_valence, entry)
where verb - the predicate lemma in the dictionary;
entry - the position of actants relative to the predicate (0 - only
the sequence subject - action - object; 1 - any sequence of subject
/ object; 2 - only the sequence object - action - subject);
obj_valence - possible object valencies, includes the structure:
  obj_valence (
  before (
    type,
    mandatory,
    tokens
    ),
case
)
    where before – the sequence of words (prepositions) to actant,
may be absent,
    type – the type of the sequence (preposition or set of words),
    mandatory – obligatory presence of a preposition before the
construction
    tokens – sets of tokens in sequence,
    case – case with valence;
add_valence – additional object valencies, similar in structure
to the obj_valence structure, with the exception of adding the
type of additional valence.
```

To clarify the structure, consider Example 3: description of the valence of the verb “contain.”

The valency of the object (obj) is represented by the accusative case Acc, and the nominative case Nom is added because of the possibility of a false definition of the case by the analyzer; however, there are no prepositions before the actant (structure before = none). Additional valencies (add) contain the valence with the locative case Loc and the obligatory preposition “on” or “at” in front of the actant.

The additional valence generates the auxiliary type ON (see all options in **Table 5**).

Accounting for additional valencies is necessary to eliminate the false definition of the subject and object. In this case, the object is defined by the nominative case in full-length segments and the anchor point in the subordinate segments.

The process of predicate-argument structures extraction in a generalized form is presented in the activity diagram (see **Figure 6**).

When detected in the segment of the target verbs (see **Table 2**), the sequence of currents to the right and left of the verb extends to the left and right parts, respectively. For full-length segments (types “N_V_N,” “N_which_V_N,” “which_V_N,” and “N _-_ N” from **Table 1**), the right and left parts are formed in the same segments, and for subordinates (type “V_N”), the left part is replaced and forms the found subject connecting.

Next comes the key stage of extraction: the search for the structures of the object and subject in the selected parts of the segment according to the predicate valencies. The order of searching for a subject and an object is determined by the possible position of actants indicated in valency (the entry field). The sequence of determining the actants in the parts of the segment for any position of the subject and object is presented in **Figure 7**.

The search for valencies in the definition of a subject and an object in parts of a segment is carried out using the token-marking mechanism. The following label sets are used:

- For the subject (P-SBJ, N-SBJ, I-SBJ, and A-SBJ)
- For an object (P-OBJ, N-OBJ, and I-OBJ)
- For auxiliary valencies (P-ON, P-VIA, N-ON, N-VIA, N-BY, N-WHOM)

The label prefix corresponds to the functional purpose of the token: preposition (P); noun (N), and adjective (A), the top of the noun group; and inside (I), the entry of a token into a noun group.

Type of valence	Preposition before actant	Case of actant
ON	On/at	Prepositional case
VIA	Through	Accusative case
BY	—	Instrumental case
WHOM	—	Accusative case

Table 5.
Generated types of additional valencies of verbs.

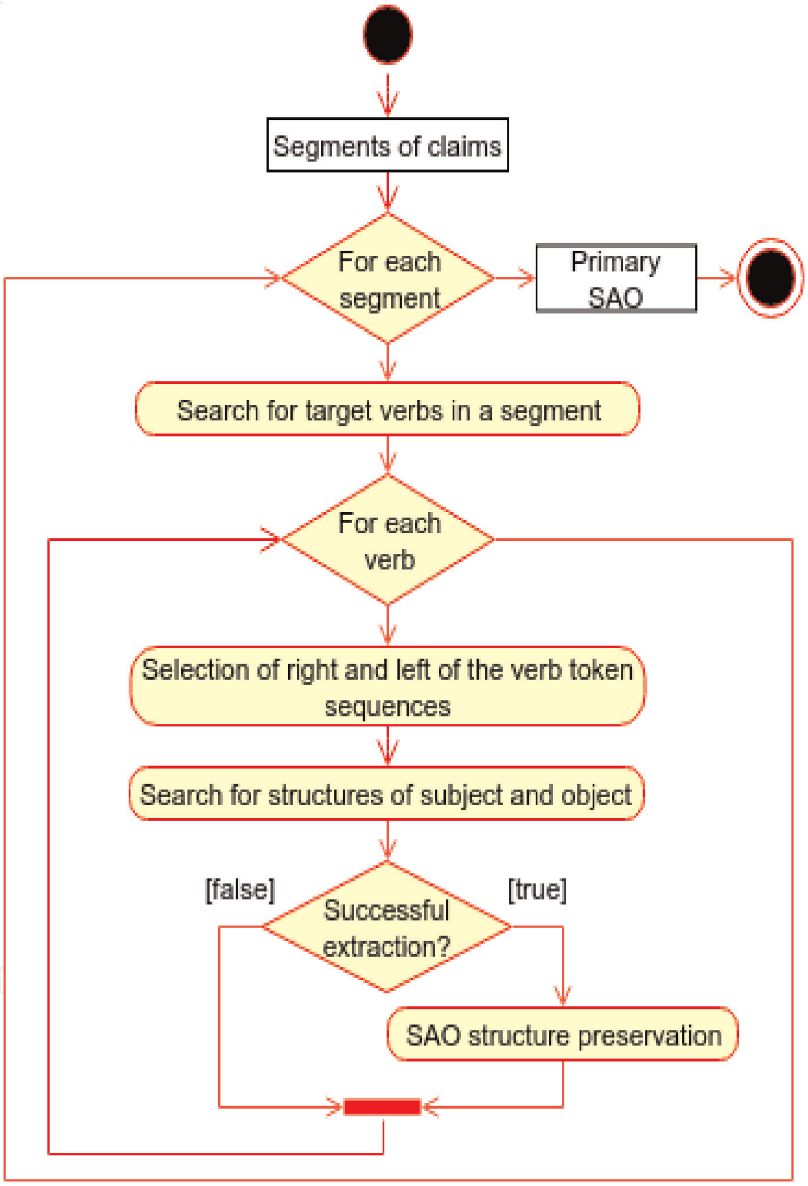

Segments of claims
For each segment
Primary SAO
Search for target verbs in a segment
For each verb
Selection of right and left of the verb token sequences
Search for structures of subject and object
[false]
Successful extraction?
[true]
SAO structure preservation

Figure 6.
The general algorithm for extracting SAO structures.

Marks in parts of the segment are placed for the object according to the predicate valences and for the subject, for the nominal case of a noun (or less often an adjective). If there is a mandatory preposition in valency, the occurrences of the specified preposition are first sought; in case of failure, the search is interrupted. The search for additional valences for an object is more priority than the search for the valence of the object itself, i.e., starts first.

Let us explain the presented methodology of SAO extraction on the example of a phrase: “[gasket] contains [at the other end one element]” (“[прокладка] содержит

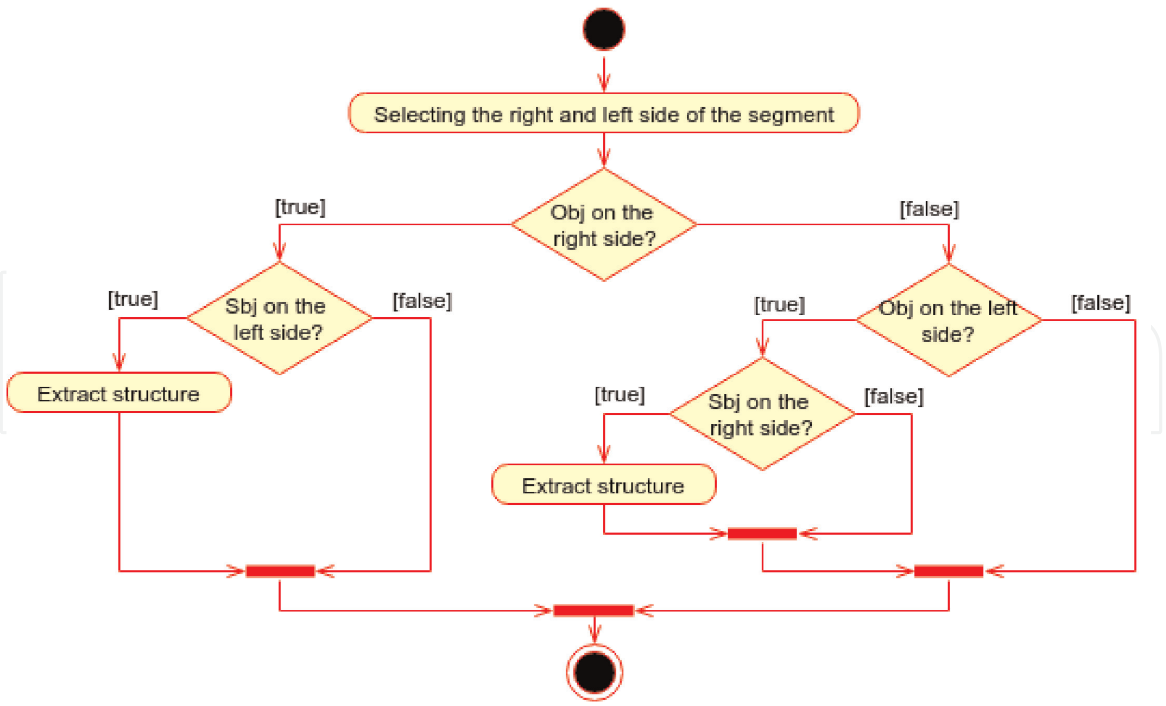


Figure 7.
The sequence of defining the actants.

Token	Lemma	POS
“gasket” (“прокладка” in Russian)	прокладка	NOUN
“contains” (“содержит” in Russian)	содержать	VERB
“at” (“на” in Russian)	на	ADP
“other” (“другом” in Russian)	другой	ADJ
“end” (“конце” in Russian)	конец	NOUN
“on” (“один” in Russian)	один	NUM
“element” (“элемент” in Russian)	элемент	NOUN

Table 6.
Morphology for each token.

на [другом конце один элемент]” in Russian). In this case, brackets indicate the boundaries of noun groups. A brief morphological analysis for each token is presented in **Table 6**.

Step 1. Search for target verbs

The target verb is the “contain” token, the valencies of which are described in Example 3.

Step 2: Select the right and left sequence of tokens from the verb

Left side (left part, l_part): “gasket”

Right part (right part, r_part): “one element at the other end”

Step 3: Search for structures of subject and object

A single sequence of subject and object is possible relative to the predicate “contain”: subject-action-object (the entry field). Therefore, the search object is carried out only in the right part (r_part).

Step 3.1: Search for additional valencies

The predicate has additional valencies (general search pattern: [“on/at” Noun < Loc>]), for the object they are searched for first.

There is a search for a mandatory preposition “on/at,” it is present on the right side of the segment and is marked (1). After that, the labeling of the additional valence continues: it is necessary to find a noun with the prepositional (Loc); it is also present (the token “end”) and marked (2). Since the valence is actually found, it is necessary to label homogeneous members (from the preposition to the noun within the noun) with a marker with the prefix “I” (3).

Final marking after searching for additional valencies:

Token	At	The other	End	One	Element
Marker	P-ON (1)	I-ON (3)	N-ON (2)	—	—

Step 3.2 Search for object valence

The valency of the object contains only one position:

```
'obj':[{
     'before':None,
     'case':[ 'Acc','Nom'],
}]
```

In this case, before the actant, there are no prepositions. It is enough to find a noun, pronoun, or adjective in the necessary case (accusative (Acc) or nominative (Nom) case); and the accusative case will be sought first since it is listed first. Such an anchor point is the token “element,” and it is marked with the label of the vertex of the noun group N-OBJ (1). As valency is found, it is necessary to complete the marking of a noun phrase of the vertex. In the sentence, there is the only token “one” (the marker of a noun phrase of a token “one” (2) matches a marker of a noun phrase of vertex “element”). Other markers (additional valencies) are not rewritten.

Final marking after the search of a valency of an object:

Token	at	the other	end	one	element
Marker	P-ON	I-ON	N-ON	I-OBJ (2)	N-OBJ (1)

Thus, an object is found, and it is possible to pass to the search for the subject.

Step 3.3 Search for subject valency

Subject search is carried out in the remaining left part. Since this is a full segment, as a subject it is necessary to find a noun, pronoun, or numeral in the nominative case. In the left part, there is only one token, “gasket,” and it satisfies the search criteria, marked as N-SBJ.

Final marking of the right side of the segment:

Step 3.3 Search for subject valency

Subject search is carried out in the remaining left part. Since this is a full segment, as a subject it is necessary to find a noun, pronoun or numeral in the nominative case. In the left part there is only one token - “gasket”, and it satisfies the search criteria; marked as N-SBJ.

Final marking of the right side of the segment:

Token	gasket
Marker	N-SBJ

Subject	"gasket" [N-SBJ]
Action	contain
Object	"at the other end **one element**" → "one element" [P-ON I-ON N-ON **I-OBJ N-OBJ**]

Step 4. Formation of predicate-argument construction

Since the extraction of the subject and the object was carried out successfully, the formation of the triplet as an SAO follows; noun groups of the subject and object are distinguished by labeling without taking into account additional valences.

This structure is used as an intermediate form of data processing. In the future, the primary SAOs are subjected to processing aimed at clarifying the structural elements and linking them together.

3.3 Creation of a tree of relations of elements of structures

The creation of a tree of relationships requires determining the final vertices of the graph, taking into account the presence of identical concepts, as well as identifying implicit relations of generic terms between elements.

For further binding of the primary SAOs, preprocessing is necessary:

- The separation of homogeneous members in the description of structural elements along the border of the composing union "and" provided it is located between the labels of the vertices of the noun group (i.e., "N-SBJ" or "N-OBJ").

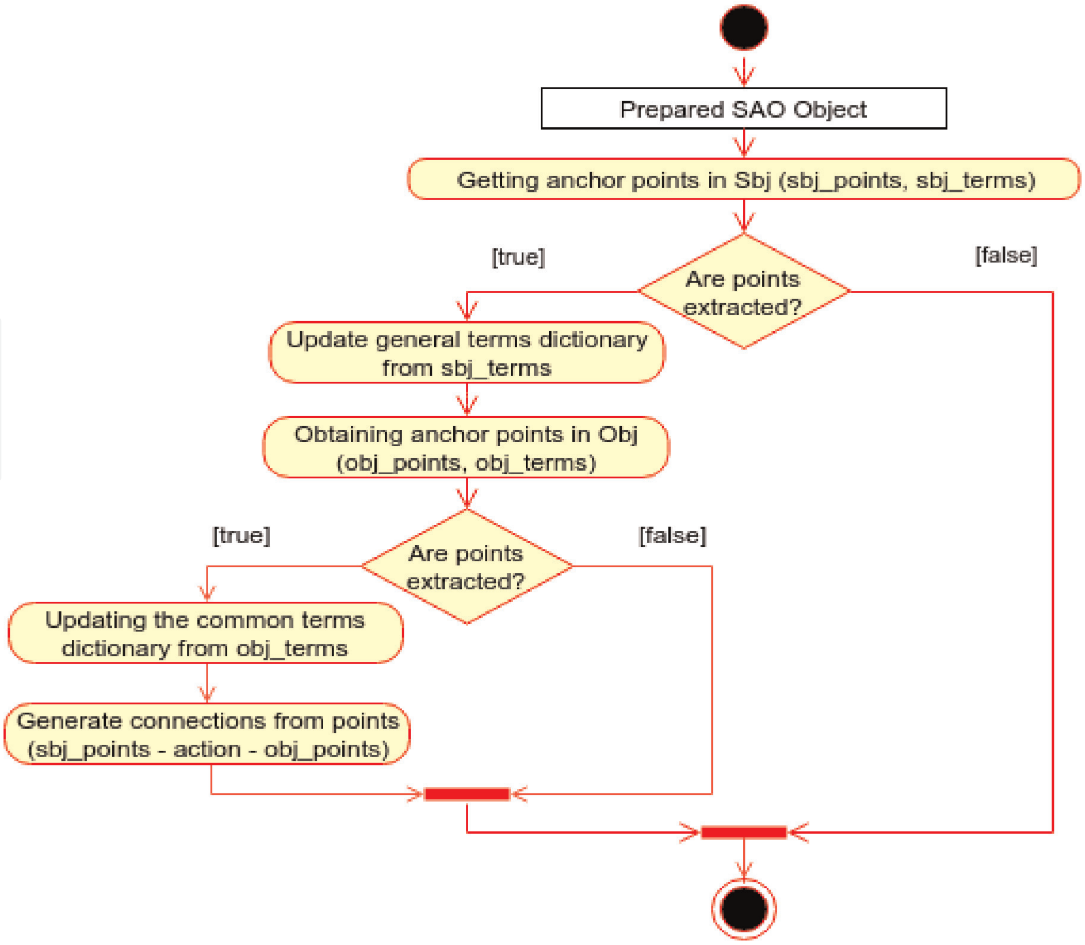

Prepared SAO Object
Getting anchor points in Sbj (sbj_points, sbj_terms)
[true]
[false]
Are points extracted?
Update general terms dictionary from sbj_terms
Obtaining anchor points in Obj (obj_points, obj_terms)
[true]
[false]
Are points extracted?
Updating the common terms dictionary from obj_terms
Generate connections from points (sbj_points - action - obj_points)

Figure 8.
The general sequence of building links of actants.

- Generation of the normal form of the actants (in the nominative case) and the genitive case (further as the Gent-form).

- After preprocessing, the primary SAO is considered prepared.

When saving an actant, it is necessary to form its normal form (the nominative case), which will be saved as a structural element. For this, the morphological characteristics of the top of the actant are determined, and all previous tokens to the top are put in agreement in the form of the nominative case. For example, "(to) hollow cylinders of stator and rotor" → "*hollow cylinders of rotor and stator*" ("(к) полым цилиндрам статора и ротора" → "*полые цилиндры* статора и ротора" in Russian).

Similar occurs and the declination of the actant is in the form of the genitive case. Gent-form is necessary for finding the parental relationship between the actants, for example, "(with) first output" → "*first output*" ("(с) первым выходом" → "*первого выхода*" in Russian).

Actants are declined using the pymorphy2 tool.

This is followed by linking the prepared SAOs into a single graph. The mechanism consists in the sequential transformation of actants of subjects and objects into a set of anchor points from a common glossary of terms, followed by memorization of the relation (predicate) for the identified points of the subject and object. The overall sequence of building links is represented by the activity diagram in **Figure 8**.

The general graph of relations is stored in the form of a dictionary of terms, a dictionary of relations, and a map of relations containing the parental relations between the actants through predicates (from the dictionary of relations).

Anchor points are formed when analyzing the subject and object of the prepared SAOs. When terms are added (the normal form of an actant's sentence), the parent relationship with another term is searched. The operation is carried out under the condition that the normal form of the actant has a sequence of tokens in the genitive case (marked with an "I-GEN" label in the analysis). An example of a parent relationship search is shown schematically in **Figure 9**.

If the final sequence of tokens in the processed actant is in the genitive case (1), then the word forms in the array of prepared SAO (2) are checked for coincidence. The parent actant index in the terms vocabulary (3) is restored by the bundle identifier or normal form. The processed actant is cleared from the gent-ending, and a new actant is added to the terms vocabulary (5). The "have" parent

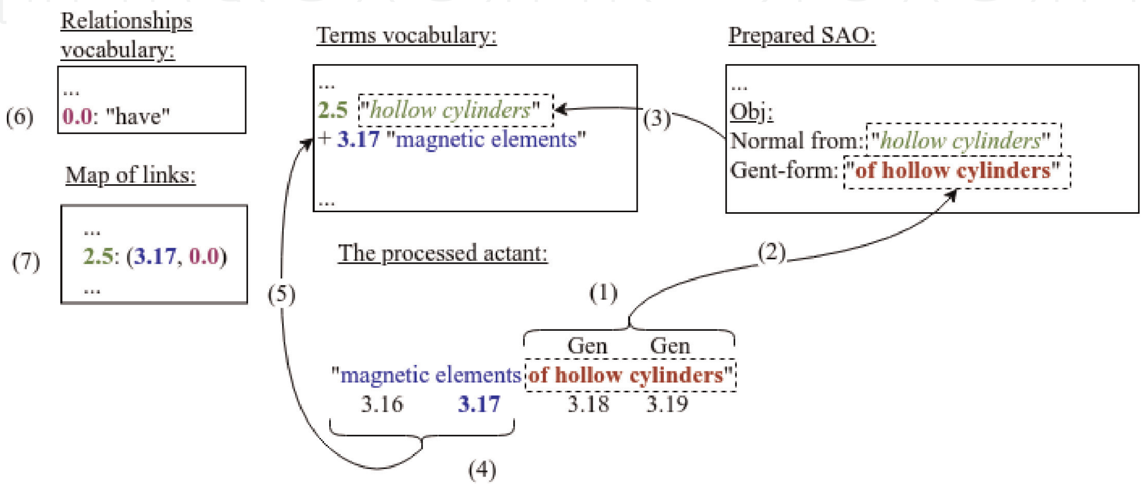


Figure 9.
The search logic of the parent relationship.

relationship is already included by default in the relationship vocabulary (6). Next, the map of links is updated to reflect the new ratio of anchor points (7).

In this case, the indices in the vocabularies are the coordinates of the vertex of the noun group (including the segment number and the token number in the segment).

After receiving the anchor points, consisting of the coordinates and the normal form of the structural elements, the map of connections is filled: the actants of the subject act as parent concepts for the actants of the object connected through the predicate. However, for the semantic class of predicates of relations (e.g., “connect,” etc., see **Table 2**), the direction is not so relevant, since it is mutual. The results of processing the prepared SAOs are buffered, and after checking the semantic classes of actants, the graphs of the entire device are updated.

To consider the complete structure of the output, take a small fragment of the text modeling the description of the claims:

“Magnetic gearbox, characterized in that the hollow cylinders are connected with the rotor of slow rotation and with the stator, and the magnetic elements of the hollow cylinders have an angular position” (“Магнитный редуктор, отличающийся тем, что полые цилиндры связаны с ротором медленного вращения и со статором, а магнитные элементы полых цилиндров имеют угловое положение” in Russian).

Primary predicate-argument constructs will be represented by the following SAOs:

Prepared SAO number 1:

Subject	Tokens	hollow cylinders (“полые цилиндры” in Russian)
	Markers	[“I-SBJ,” “N-SBJ”]
Action	are connected (“связанный” in Russian	
Object	Tokens	with the rotor of slow rotation and with the stator (“с ротором медленного вращения и со статором” in Russian)
	Markers	[“P-OBJ,” “N-OBJ,” “I-OBJ,” “I-OBJ,” “I-OBJ,” “P-OBJ,” “N-OBJ”]

Prepared SAO number 2:

Subject	Tokens	magnetic elements of the hollow cylinders (“магнитные элементы полых цилиндров” in Russian)
	Markers	[“I-SBJ,” “N-SBJ,” “I-GEN,” “I-GEN”]
Action	have (“имеют” in Russian)	
Object	Tokens	an angular position (“угловое положение” in Russian)
	Markers	[“I-OBJ,” “N-OBJ”]

The constructed graph is presented in **Figure 10**.

From example, it can be noted that homogeneous members are distinguished from the object of the bundle SAO No. 1 and the parent relationship of “magnetic elements” (“магнитных элементов” in Russian) and “hollow cylinders” (“полых цилиндров” in Russian) is expressed by the pseudo-relationship “have.” In this case, the actant “angular position” (“угловое положение” in Russian) was not added to the output set, since it falls into the forbidden semantic class of words.

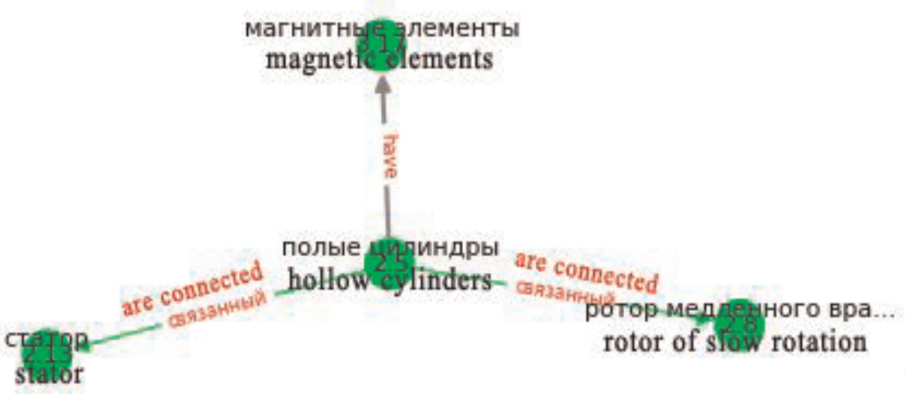


Figure 10.
Constructed graph of structural elements.

4. Construction of domain ontology

The ontology-based patent processing technologies are developing more and more actively. Thus, in [7], ontology extracted concepts and relationships are used to improve patent search. In [29] the information extracted from the claims of the device is stored in an ontological representation and is used for visualization and processing. In [30] patent information in ontology is formalized with reference to technological areas. Thus, the ontological representation provides ample opportunities for the implementation of the description and linking and searching for patent information.

In this paper, at the initial stage, ontology is considered to a greater extent as a storage medium. Inventions and connections between them act as concepts. The designed scheme of the domain ontology is presented in **Figure 11**.

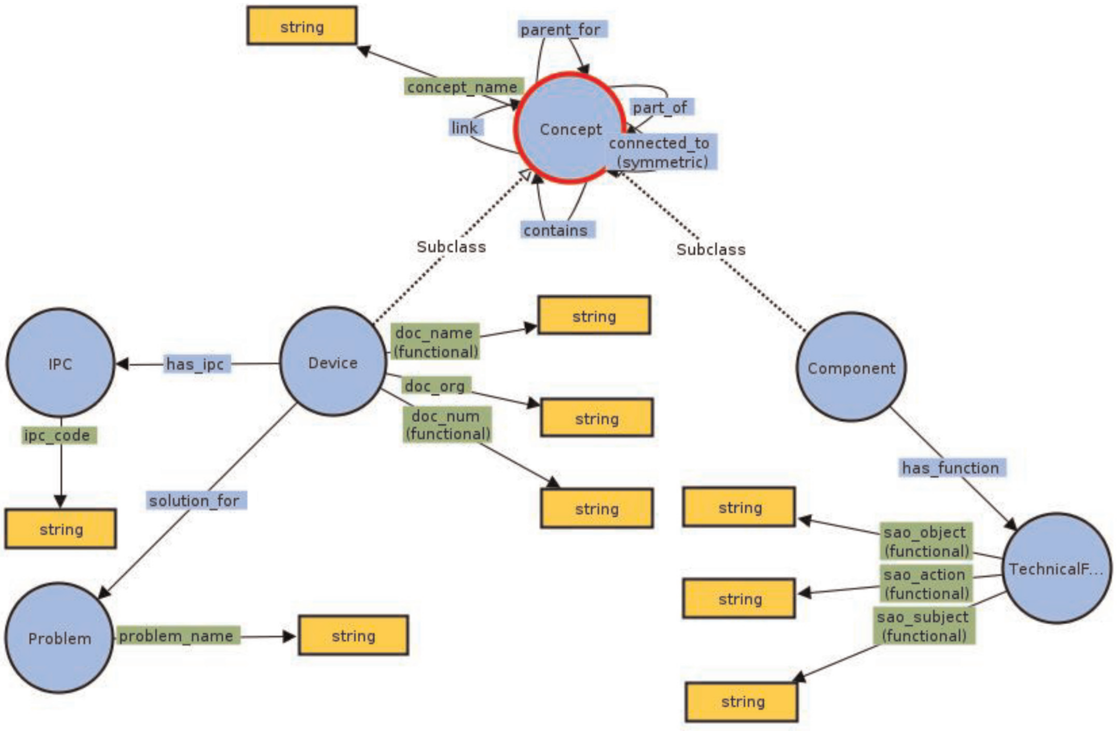
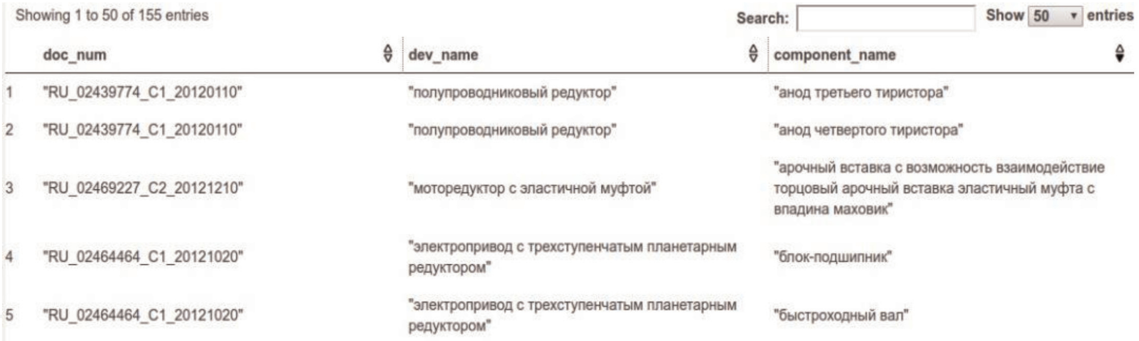


Figure 11.
Ontology scheme for storing patent information.



Showing 1 to 50 of 155 entries

	doc_num	dev_name	component_name
1	"RU_02439774_C1_20120110"	"полупроводниковый редуктор"	"анод третьего тиристора"
2	"RU_02439774_C1_20120110"	"полупроводниковый редуктор"	"анод четвертого тиристора"
3	"RU_02469227_C2_20121210"	"моторедуктор с эластичной муфтой"	"арочный вставка с возможность взаимодействие торцовый арочный вставка эластичный муфта с впадина маховик"
4	"RU_02464464_C1_20121020"	"электропривод с трехступенчатым планетарным редуктором"	"блок-подшипник"
5	"RU_02464464_C1_20121020"	"электропривод с трехступенчатым планетарным редуктором"	"быстроходный вал"

Figure 12.
The output of all components of a given invention.

The invention (patent document) is assigned with the name of the invention, the patent number, the owner organization, and the IPC codes. Additional concepts of the technical function and the problem solved by the invention are introduced for the subsequent development of the system.

Relationships between components are specified through the following properties:

- connected_to—connection between elements (verbs “set,” “connect,” etc.)
- contains—indicating the presence of a component (verbs “contain,” “have,” etc.)
- part_of—an implicit indication of the relationship to the invention
- parent_for—an indication of an implicit parental relationship between the elements (e.g., “hollow stator cylinders”)

The ontology graph is described in RDF/XML format. The extracted semantic structures are embedded in the description of ontologies and are uploaded as an OWL file. The resulting data can then be loaded into the RDF storage and make the appropriate SPARQL queries. The data processing processed by the system amounted to 11,200 patent documents.

Further, it is possible to make various requests for information retrieval. An example of a simple request for the conclusion of all the components of a given invention (by the entry of the word “reducer” in the name) is as follows:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX cad: <http://www.vstu.ru/cad/ontology-of-devices#>
SELECT ?device ?dev_name ?component_name
WHERE{
 ?device rdf:type cad:Device .
 ?device cad:concept_name ?dev_name .
 ?device cad:doc_name ?doc_name .
 ?device cad:doc_num ?doc_num .
 ?component cad:part_of ?device .
 ?component cad:concept_name ?component_name
 FILTER(CONTAINS(STR(?dev_name), "редуктор"))
}
```

The result of the query is shown in **Figure 12**.

You can see which inventions use the given component or search for the descendants, etc. At the same time, unlike relational databases, the ontological representation allows you to add descriptive logic and, in general, make more flexible queries to data.

5. System evaluation methodology

The quality of data extraction was assessed for primary SAOs (i.e., excluding post-processing with the separation of homogeneous members and taking into account the semantic classes of actants). For this purpose, manual marking of an independent test set was carried out, including the main points of the claims of 30 patents. The markup was made taking into account the semantic classes of verbs introduced into the system, which characterize the target relations between the components of the technical object (see **Table 2**). In this case, it was allowed to add into the system descriptions (valencies) of previously not encountered target verbs.

The following metrics were used as evaluation metrics: precision, recall, and F1 measure.

Precision is the proportion of correctly extracted engineering implementations (SAO) to the total number of extracted engineering implementations, defined by formula (1):

$$Precision = \frac{|R_{rel}|}{|R_f|} \tag{1}$$

where R_{rel} is the relevant SAO and R_f are all SAOs extracted by the system.

Recall is the ratio of the number of correctly extracted engineering implementations to the total number of relevant structures, defined by formula (2):

$$Recall = \frac{|R_{rel}|}{|S_{rel}|} \tag{2}$$

where R_{rel} is the relevant SAO and S_{rel} are manually selected SAOs.

F1 measure is the harmonic mean of the accuracy and recall; it is determined by formula (3):

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{3}$$

where precision is the extraction accuracy and recall is the extraction completeness.

Parameter	R_f	R_{rel_1}	R_{rel_2}	S_{rel}
Amount	305	198	248	318

Table 7.
Extracted test data set objects.

Parameter	R_f	R_{rel_1}	R_{rel_2}	S_{rel}
Amount	142	118	129	136

Table 8.
Extracted design data set objects.

Since the unit of extraction is a predicate-argument construct, there may be cases of incomplete extraction of one (or both) of its arguments (due to the erroneous identification of the noun group and other reasons). Therefore, two types of evaluation were introduced:

- With a rigorous entry of arguments (there must be a complete occurrence of the selected words)

- With a lax entry of arguments (only the hit of the vertex of the noun group is significant)

Example of mild occurrence of arguments: “the second node is connected to *another* output of the primary winding”:
S: the second node
A: is connected
O: to output of the primary winding

In the noun group of the object, the definition of “another” is omitted, and the vertex of the nominal group “output” is present. Therefore, according to the evaluation with a rigorous entry, the structure is considered to be incorrectly recognized. However, by a lax evaluation, the extraction can be considered successful.

Metrics	**Test data set**		**Design data set**	
	Lax evaluation	**Rigorous evaluation**	**Lax evaluation**	**Rigorous evaluation**
Precision	0.81	0.65	0.91	0.83
Recall	0.78	0.62	0.95	0.87
F1 measure	0.79	0.63	0.93	0.85

Table 9.
Metric counting results.

Configuration number	**Operating system**	**Device hardware configuration**
1	CentOS Linux (7.3.1611) × 64	Intel Xeon CPU E5-2650 v3 @ 2.30 GHz, RAM 132 Gb
2	Ubuntu Linux (18.04) × 64	AMD A8 6410 @ 2GHz, RAM 8 Gb

Table 10.
Hardware configuration.

Program pass	**Hardware configuration #1 (cluster)**	**Hardware configuration #2 (laptop)**
Pass 1	2241.3	5614.24
Pass 2	2161.8	5602.21
Pass 3	2151.1	5604.37
Average processing time, sec	2180	5606.94
Average processing time per document, sec	0.581	1.49

Table 11.
The results of the evaluation of the system speed.

An additional evaluation of the quality was also carried out for the design data set of the claims (i.e., on which the system was designed).

A test data set of 30 documents on the basis of manual marking included 318 SAOs. The results of counting the number of extracted test data set objects are presented in **Table 7**.

Designations of headings (according to formulas (1) and (2)):

- R_f—all SAO structures extracted by the system
- R_{rel_1}—relevant SAO structures with a rigorous entry of the entire noun group in the argument
- R_{rel_2}—relevant SAO structures taking into account that the vertex of the noun group is in the argument
- S_{rel}—manually selected SAO structures

The design data set of 14 documents contained 136 recoverable structures. The results of counting the number of extracted SAOs of the design data set are presented in **Table 8**.

The result of the counting of the extraction quality metrics (according to formulas (1)–(3)) for the test and design data set is presented in **Table 9**.

According to the results of the evaluation on the test data set, the proposed method allows to extract data with an accuracy of 63%.

The performance indicator was the average patent processing time (reading from an XML document). The speed of work was determined by the arithmetic average of the time of three runs on a test data set of 3755 documents without marking for each hardware configuration (see **Table 10**).

The results of the run time estimation are presented in **Table 11**.

Patent processing speed is on average less than 1.5 seconds, which is not difficult for batch processing.

6. Conclusion

The general task of this research was extracting information for prior art patent search and new technical solution synthesis based on the analysis of Russian-language patents. The data source was the main claim of the device from the patent text. And the object of extraction was the SAO semantic structures.

Extracting structured data from patents depends on the specifics of the writing of documents. The peculiarities of writing claims of the invention include the excessive length of sentences, the complication of the participle clauses, and characteristic terminology. Practical experience has shown that NLP tools do not always produce a stable result for the patent processing. Thus, they are ineffective in building high-quality data extraction systems with direct application. Therefore, a special approach to the preprocessing of patent texts seems necessary. It is worth noting that in practice they often use a combination of tools to improve the quality of processing.

Due to the sublanguage of patents and the availability of common templates for the formation of documents, the author made an assumption about the sufficiency of morphological and shallow parsing analysis with a number of heuristics to effectively extract structural elements from the description of the claims.

The developed method of extracting SAO structures involves two main stages: the segmentation of the sentences of the claims and the extraction of technical implementations in the form of predicate-argument structures. Segmentation is based on the derivation of typical structures in the sentence and coordination of dependent words on cases. The extraction of structural elements of technical objects is based on a verb set corresponding to certain semantic classes. After that, the extracted structural elements are linked to a graph, taking into account homogeneous members and parental relations.

The effectiveness of the method was evaluated with an independent test data set of patents with a total number of marked SAO structures of 318 elements. The value of the F1 metric with a rigorous evaluation (full comparison of arguments) and a lax evaluation (the presence of vertexes of noun groups suffices) was 63 and 79%, respectively. The average document processing speed was 1.49 seconds on a laptop with an average configuration.

The precision and recall of Russian-language patent processing exceed the results described in the article [1]. However, not the most advanced level of data extraction of 63% with a rigorous evaluation is due to the inevitable errors of the tools and to the imperfection of the embedded processing logic. It is impossible to take into account all the subtleties of the formation of the sentences especially for a rich Russian language with free word order. The assumption of the finiteness of the segment patterns in the claims is just too strong. However, with a more detailed study, it seems possible for the authors to cover most of the typical variants of writing segments of the claims and bring the system to an industrial level. The results show the prospects of the proposed approach.

The explicit description of many extraction mechanisms (heuristics) hampers the development of the system and the coverage of all possible variations of writing patents. However, the method can still be considered extensible. The inherent stereotyped patent allows not resorting to overly difficult decisions and maintaining a balance between the complexity of the system and its effectiveness.

Further tasks include a more detailed selection of text processing tools, identification of effective methods for statistical extraction of segment patterns, development of heuristics for extracting SAO structures, and improved graph building logic.

The construction of the graph of structural elements of a technical object allows going on to compile the ontology of the subject area and come closer to solving the problems of prior art patent search and information support for new technical solutions synthesis, which is a further direction of the research.

The ontology scheme as a concept includes the structural elements of technical objects and the relationship between them, as well as supporting information on the invention. The initial content of the ontology is based on the processing of 11,200 patent documents for inventions. The existing scheme already allows retrieving useful information about alternatives of structural components and communications between them, for example, searching for all elements of a structure in a given invention or tracking relationships. The results suggest that the proposed approach is promising. A further direction of research is seen by the authors in improving the existing method for extracting data and expanding ontology.

Acknowledgements

The reported study was funded by RFBR (research project 18-07-01086), RFBR, and Administration of the Volgograd region (projects 19-47-340007 and 19-41-340016).

Author details

Dmitriy Korobkin*, Sergey Vasiliev, Sergey Fomenkov and S.G. Kolesnikov
Volgograd State Technical University, Russia

*Address all correspondence to: dkorobkin80@mail.ru

References

[1] Korobkin DM, Fomenkov SA, Kravets AG, Kolesnikov SG. Methods of statistical and semantic patent analysis. In: Kravets A, Shcherbakov M, Kultsova M, Groumpos P. Volgograd State Technical University, et al., editors. CIT and DS 2017: Proceedings. Germany: Springer International Publishing AG; 2017. pp. 48-61. (Ser. Communications in Computer and Information Science; Vol. 754)

[2] Mel'čuk IA. Dependency Syntax: Theory and Practice. NY: SUNY Publ; 1988

[3] Choi S et al. SAO network analysis of patents for technology trends identification: A case study of polymer electrolyte membrane technology in proton exchange membrane fuel cells. Scientometrics. 2011:863-883. DOI: 10.1007/s11192-011-0420-z

[4] Guo J, Wang X, Li Q, Zhu D, et al. Subject–action–object-based morphology analysis for determining the direction of technological change. Technological Forecasting and Social Change. 2016;**105**:27-40

[5] Wang X et al. Identification of technology development trends based on subject–action–object analysis: The case of dye-sensitized solar cells. Technological Forecasting and Social Change. 2015;**98**:24-46

[6] Yang C et al. SAO semantic information identification for text mining. International Journal of Computational Intelligence Systems. 2017;**10**:593-604. DOI: 10.2991/ijcis.2017.10.1.40

[7] Phan C-P, Nguyen H-Q, Nguyen T-T. Ontology-based heuristic patent search. International Journal of Web Information Systems. 2018. DOI: 10.1108/IJWIS-06-2018-0053

[8] Souili A, Cavallucci D, Rousselot F, Zanni-Merk C. Starting from patents to find inputs to the problem graph model of IDM-TRIZ. Procedia Engineering. 2015;**131**:150-161. DOI: 10.1016/j.proeng.2015.12.365

[9] Supotnitskiy MV. Formula izobreteniya. Vedomosti Nauchnogo tsentra ekspertizy sredstv meditsinskogo primeneniya. 2013;**1**: 41-44

[10] Tomita-parser Developer Guide. Available from: https://tech.yandex.ru/tomita/doc/dg/concept/about-docpage/ [Accessed: 10 May 2019]

[11] Rubaylo AV, Kosenko MYU. Programmnyye sredstva izvlecheniya informatsii iz tekstov na yestestvennom yazyke. Al'manakh sovremennoy nauki i obrazovaniya. 2016;**12**:87-92

[12] Suleymanov RS. Izvlecheniye metadannykh iz polnotekstovykh elektronnykh russkoyazychnykh izdaniy pri pomoshchi Tomita-parsera. Programmnyye produkty i sistemy. 2016;**4**(116):58-62

[13] Koblikov IA, Korobkin DM, Fomenkov SA, Yarovenko VA. Metodika izvlecheniya opisaniy realizuyemykh v patente tekhnicheskikh funktsiy. Izvestiya Volgogradskogo gosudarstvennogo tekhnicheskogo universiteta. 2017; **8**(203):55-59

[14] Link Grammar Parser Homepage. Available from: http://www.abisource.com/projects/linkgrammar [Accessed: 10 May 2019]

[15] MaltParser Homepage. Available from: http://maltparser.org/ [Accessed: 10 May 2019]

[16] UFAL UDPipe Homepage. Available from: http://ufal.mff.cuni.cz/udpipe [Accessed: 10 May 2019]

[17] Korobkin DM, Tyulkina EA, Fomenkov SA, Kolesnikov SG. Sistema izvlecheniya tekhnicheskikh funktsiy iz patentnogo massiva. ITNOU: Informatsionnyye tekhnologii v nauke, obrazovanii i upravlenii. 2017;**2**:24-30

[18] UD Russian SynTagRus Treebank's page. Available from: https://universaldependencies.org/treebanks/ru_syntagrus/index.html [Accessed: 10 May 2019]

[19] CoNLL-U Format. Available from: https://universaldependencies.org/format.html

[20] Smirnov IV, Salmanov AO, Kuznetsova ES, Haramain IV. Semantiko-sintaksicheskiy analiz yestestvennykh yazykov Chast' II. Metod semantiko-sintaksicheskogo analiza tekstov. Iskusstvennyy intellekt i prinyatiye resheniy, ISA RAN. 2014;**1**: 11-24

[21] Vasiliev SS, Kharitonov AA, Korobkin DM, Fomenkov SA. Extracting descriptions of morphological features of technical objects from Russian-language patents. Modeling, Optimization and Information Technologies. 2018;**4**:6

[22] Kobzareva TYU. V poiskakh sintaksicheskoy struktury: avtomaticheskiy analiz russkogo predlozheniya s oporoy na segmentatsiyu. Moscow: RGGU; 2015

[23] Kharlamov AA, Ermolenko TV, Dorokhina GV. Sravnitel'nyy analiz organizatsii sistem sintaksicheskikh parserov. Inzhenernyy vestnik Dona. 2013;**4**(27):74

[24] Asiryan AK. Morphological tagging tools comparison. In: Intellectual Potential of the XXI Century. 2017. Available from: https://www.sworld.com.ua/konferu7-317/27.pdf [Accessed: 10 May 2019]

[25] Dereza OV, Kayutenko DA, Fenogenova AS. Automatic morphological analysis for Russian: A comparative study. In: Computational Linguistics and Intellectual Technologies: Proceedings of the International Conference "Dialogue". 2016. Available from: http://www.dialog-21.ru/media/3473/dereza.pdf [Accessed: 10 May 2019]

[26] Kuzmenko E. Morphological analysis for Russian: Integration and comparison of taggers. In: Ignatov D et al., editors. Analysis of Images, Social Networks and Texts. AIST 2016. Communications in Computer and Information Science, Vol 661. Cham: Springer; 2017. pp. 162-171

[27] Korobov M. Morphological analyzer and generator for Russian and Ukrainian languages. Analysis of Images, Social Networks and Texts. 2015:320-332. arXiv:1503.07283v1

[28] Fenogenova A. Chanker imennykh grupp russkogo yazyka. Available from: http://web-corpora.net/wsgi/chunker.wsgi/npchunker/npchunker/ [Accessed: 10 May 2019]

[29] Reis SRN, Reis A, Carrabina J, Casanovas P. Contributions to modeling patent claims when representing patent knowledge. Lecture Notes in Computer Science. 2018;**10791**:140-156. DOI: 10.1007/978-3-030-00178-0_9

[30] Ulmschneider K, Glimm B. Semantic exploitation of implicit patent information. In: Proceedings of the 2016 IEEE Symposium Series on Computational Intelligence (SSCI 2016). Athens, Greece; 2016. DOI:10.1109/SSCI.2016.7849943