

METHOD OF INFORMATION OBTAINING FROM ONTOLOGY ON THE BASIS OF A NATURAL LANGUAGE PHRASE ANALYSIS

Anna Litvin^{a[0000-0002-5648-9074]}, *Vitalii Velychko*^{a[0000-0002-7155-9202]}, *Vladyslav Kavarynskyi*^{b[0000-0002-6940-579X]}

^a V.M. Glushkov Institute of Cybernetics of National Academy of Sciences of Ukraine, Kyiv, Ukraine

^b I.N. Frantsevich Institute of Problems of Materials Science of National Academy of Sciences of Ukraine

Разработан метод анализа фраз на естественных языках флективного типа (украинский и русский), позволяющий выделить в предложениях основные идеи и группы слов, при помощи которых они излагаются. Сформированные таким образом семантические деревья высказываний, каждое из которых выражает одну конкретную идею, являются удобным исходным материалом для построения запросов к онтологии на языке SPARQL. Метод анализа предложений включает следующую последовательность основных этапов: разбиение на слова, выделение маркерных слов и словосочетаний, определение типа высказывания, выделение именных групп, составление синтаксического графа предложения, построение семантических деревьев высказываний, основанных на имеющихся типах высказываний, подстановка параметров из семантических деревьев высказываний в соответствующие шаблоны SPARQL запросов. В статье приведена UML-диаграмма деятельности системы, которая реализует описанную модель, и примеры шаблонов SPARQL запросов. Выбор соответствующего шаблона запроса зависит от типа высказывания, выраженного данным семантическим деревом высказывания. Понятия, полученные в качестве ответа на запрос, связываются с соответствующим семантическим деревом высказывания. В случае неполучения информации из онтологии, производится редукция именных групп для выражения более общих понятий и построение запросов с их использованием. Это позволяет всегда получить некоторый ответ, хотя и не столь точный, как при использовании полной именной группы. Использование шаблонов SPARQL запросов требует априорно заданной структуры онтологии, которая также предлагается в данной работе. Такая система применима для ведения диалога с помощью чат-бота, для автоматического получения ответов на вопросы к тексту. Дальнейшими направлениями исследований являются расширение классификации высказываний пользователя и моделирование построения уточняющих вопросов при возникновении неразрешимых неоднозначностей.

Ключевые слова: онтология, SPARQL, анализ текста, именная группа, синтаксический граф, семантическое дерево высказывания, NLP, NLU.

Розроблено метод аналізу природно-мовних речень для мов флективного типу (українська та російська), який дозволяє виділити в реченні основні висловлені ідеї та групи слів, за допомогою яких вони викладаються. Сформовані таким чином семантичні дерева висловлювань, кожне з яких виражає одну конкретну ідею, є зручним вихідним матеріалом для побудови запитів до онтології мовою SPARQL. Метод аналізу речень включає наступну послідовність основних етапів: розбиття на слова, виділення маркерних слів і словосполучень, визначення типу висловлювання, виділення іменних груп, побудова синтаксичного графа речення, побудова семантичних дерев висловлювань, заснованих на наявних типах висловлювань, підстановка параметрів з семантичних дерев висловлювань до відповідних шаблонів SPARQL запитів. У статті наведено UML-діаграма діяльності системи, яка реалізує описану модель, і приклади шаблонів SPARQL запитів. Вибір відповідного шаблону запиту залежить від типу висловлювання, яке виражено семантичним деревом висловлення. Отримані в якості відповіді на запит набори понять зв'язуються з відповідним семантичним деревом висловлювання. У разі неотримання інформації з онтології, проводиться редукція іменних груп для вираження більш загальних понять і побудови запитів з їх використанням. Це дозволяє завжди отримати деяку відповідь, хоч і не настільки точну, як при використанні повної іменної групи. Використання шаблонів SPARQL запитів вимагає априорно заданої структури онтології, яка також пропонується в даній роботі. Така система може бути застосована для організації діалогу з чат-ботом або для автоматичного отримання відповідей на питання до тексту. Подальшими напрямками досліджень є розширення класифікації висловлювань користувача і моделювання побудови уточнюючих питань при виникненні нерозв'язуваних неоднозначностей.

Ключові слова: онтологія, SPARQL, аналіз тексту, іменна група, синтаксичний граф, семантичне дерево висловлення, NLP, NLU.

A method for analyzing sentences in natural languages of inflected type (Ukrainian and Russian) has been developed. The method allows one to outline main expressed ideas and groups of words in the text by which they are stated. The semantic trees of sentences formed in this way, each of which expresses one specific idea, are the convenient source material for constructing queries to the ontology in the SPARQL language. The analysis algorithm is based on the following sequence of basic steps: word tokenize; determining of marker words and phrases; identifying the available type of sentence; identifying nouns groups; building a syntactic graph of a sentence; building semantic trees of sentences based on existing types of sentences; substituting parameters from semantic trees of sentences in the corresponding SPARQL query templates. The article presents the UML-activity diagram of the system that implements the described model and examples of the SPARQL query templates. The choice of an appropriate template depends on the type of sentence expressed by a given semantic tree of a sentence. The sets of concepts received as an answer are tied as corresponding answers to the previously defined semantic tree of the sentence. In the case of non-receipt of information from the ontology, the reduction of noun groups is carried out to express more general concepts and the building queries using them. This allows us to get some answers, although not as accurate as when we use the full noun group. The use of SPARQL query templates requires an a priori known ontology structure, which is also proposed in this paper. Such a system is applicable for dialogue using chat-bots or for automatically receiving answers to questions from the text. Further research directions are the expansion of the classification of the user speech and modeling of building clarifying questions when an unresolvable ambiguity.

Keywords: ontology, SPARQL, text analysis, noun group, syntactic graph, semantic tree of a proposition, NLP, NLU.

Introduction

This paper considers the application of methods for analyzing natural language texts written in languages of inflectional type, especially, Ukrainian and Russian. Language analysis is considered in the aspect of identifying the main ideas and intentions in the text, which is necessary for building queries to the knowledge base that provide relevant information which is using for responses to a user's request construction. Thus, the work is devoted to the application of NLP (natural language processing) and NLU (natural language understanding) methods to solve the problem of organizing a dialogue between a user and a knowledge base in a natural language. In addition, attention is paid to working with knowledge bases of ontological type.

The knowledge base is the central component of an intelligent expert system. Such systems are designed to search for solutions to problems in a certain area, based on a subject model created by an expert-user. Knowledge bases work in conjunction with systems for searching, extracting and analyzing information. A full-fledged knowledge base, in addition to structured information on a certain subject, should also contain a system of semantic (meaningful) information processing and inference rules that allow making automatic inferences. For such tasks, a hierarchical way of concepts and their relationships representing is suitable, which is one of the types of ontology [1].

The main input source for an expert system with a knowledge base is the data provided by a user. The most convenient form of data presentation in this case is usually natural language text. In this way, it is possible to provide a wide range of information for various subject areas. However, for unified work with data warehouses, computer and software systems use formalized query languages. For ontologies, SPARQL seems to be a well-known and proven query language [2]. Its usage is recommended by the W3C consortium, and it is one of the semantic web technologies [3]. Thus, processing of text in natural language for structured data obtaining that are necessary for SPARQL query building, becomes an urgent task.

In addition, for effective work with a knowledge base, it is required a certain structure of the ontology, which includes some restrictions on the initial information about the subject area presentation form. This allows one to unify SPARQL query templates for various situations, based on the types of statements and intentions from the source text. Methods for constructing a domain ontology are a separate broad topic. This article presents some aspects of the ontology structure adopted in the system we are developing. Practical aspects of the considered method are intelligent chatbots, expert systems, programs for text mining and automatic generation of conclusions basing on it. It should be noted that the system proposed in this work is designed to work primarily with grammatically and spelling correct text of scientific and technical style.

Analysis of modern achievements in natural language processing for dealing with ontological knowledge bases

Various approaches to building queries using as an initial data natural language text have been developed for several decades since communication in such way is convenient for users. New ideas for natural language interfaces are constantly being proposed. Not all of them turn out to be successful and really make a breakthrough. However, any new research in this area is valuable because it brings new ideas and a better understanding of what really works and what does not [4].

An important problem of expert systems with a natural language interface is their dependence on semantic grammars adapted to a specific database. At the same time, query languages are currently quite standardized (for example, SQL – for relational databases, SPARQL – for working with ontologies). This allows one to move away from binding to the internal structure of the database [4].

There are several approaches to natural language interface building. For example, in [5] it is proposed to focus on a limited set of semantically interpreted queries with an unambiguous mapping of relations, attributes and values, and to use statistical semantic analysis. Authors of [6, 7] proposed a model in which a program basing on a natural language text returns the problem description and forms the query code. In [8] a method is proposed for sentences generalization applied to syntactic parse trees. Authors of [9] proposed a conversion system that consists of three base components: 1 – conversion of a query in natural language to a query tree; 2 – interactive check of the conversion (by asking the user); 3 – SQL query formation.

In recent years appear methods based on machine learning and neural networks. For instance, a technique adopts reinforcement learning [10]. A neural network is used for translation of questions in natural language to the corresponding SQL queries. However, despite the huge data set in the training set, the model's accuracy is not very high: the execution accuracy is 59.4%, the logical form accuracy is 48.3%. The learning technique really depends on the subject. For such system implementation, a wide collection of the training set is needed, which is often difficult to provide.

Another possible difficultness of formal queries construction basing on natural language is cases of homonymy and ambiguity in user's phrases. Moreover, a user could use slang and specific terms of some subject area, words and phrases that have a specific meaning in the context which could be different from the common one. Some approach to this problem solving was proposed even in [11, 12]. It consists of clarifying questions addressed to the user with a proposition to clarify which of the options is correct for this case.

Works for the creation of natural language questions to SPARQL conversion methods are also performed. For example, LODQA system (Linked Open Data Question Answering) [13] – a program that receives a query in a natural language as an input and returns SPARQL queries as a performance result. The system consists of several modules. The

first module is responsible for syntactic analysis and for the creation of a graphic representation of the query, which is called a pseudo graphic template. Typically, template nodes correspond to noun groups, and links are dependencies between them. Also the pseudo graphic template points which of the nodes is the focus of the query i.e. what the user is to receive as an answer to the query. As the first module has generated a pseudo graphic template from the natural language query the second module turns on, which is responsible for URI and meanings search for the nodes of the pseudo graphic template. From one pseudo graphic template, it is possible several bind templates to obtain through normalization. The third module performs the search in the goal dataset for the corresponding parts taking into account the possible changes that could appear in the dataset. This module tries to generate SPARQL queries for all possible structure variations. Then the SPARQL queries are sending to the aim point, where the answers for them are to be obtained, and then these answers are returned to the user. The arguments of a query could be a primitive type. For making easier the identification of RDF-triplets the words of a sentence are lemmatized and the sufficient grammatical characteristics are matched to them. The considered LODQA system is oriented to deal with the English language only. The detailed features of its functioning are not given in [13], limiting to a brief description and examples analysis.

Thus, we can see that despite quite long existing term of the problem of natural language phrases to formal queries conversion it still remains actual and new approaches to its solution are proceeding to develop. Most of the works are devoted to constructing SQL queries basing on natural language (NL2SQL). There are rather few works on the conversion of natural language phrases to queries language into queries to ontologies and most of them do not reveal some technical details of the system realization. Most of the programs for natural language to formal query formation are oriented to English, but inflectional languages, such as Ukrainian and Russian) have rather different structures and need specific analysis methods.

The proposed model of natural language text analysis to obtain data for construction of queries to ontology

Ontology is a formalization of some area of knowledge through a conceptual scheme, i.e. linked data structure. Ontologies are stored in a way convenient for computer processing [14]. A standard storage format is OWL. For formal queries to ontology the most widespread language is SPARQL. However, for a human more comfortable and understandable will be information representation in a natural language. Therefore, the task of transition from a phrase or a set of phrases in natural language to a package of SPARQL queries becomes urgent. Information obtained from ontology for most of the cases will be reasonable to make out into natural language phrases.

Ontologies as a rule are built in a way to suit to a certain topic or subject. There could be several ontologies in a system. For selection of the most suitable ontology, a file could be used that contains the key terms for each of the ontologies. These lists of keywords are to be matched with words from input texts. That ontology will be selected to work with, for which list of keywords the maximum percent of coincidences in the analyzed text will be found.

Let us describe the sequence of a phrase in the natural language of inflective type analysis. To the languages of inflective type Ukrainian and Russian belong.

The first stage is graphematic analysis, which consists of tokenization of a text string into sentences, parts of complex sentences and separate words. For this purpose in our system are used instruments from NLTK library [15]. The words are represented as program objects also able to contain the word's characteristics, its place in the sentence and relationships with the other words. The usage of such structure for words representation simplifies the realization of morphological and syntactic analysis stages.

The second stage is the recognition of the marker words in the sentence for the determination of the sentence type. In Ukrainian and Russian languages the main criterion in the sentence type determination (statement or interrogative) in writing document is the presence of a question mark at the end of the sentence. Determination of a more narrow subtype of a phrase is performed through the presence of certain marker words. The accepted here classification of phrases is based on [1]. In our model for the moment the following simple subtypes are defined: "neutral narration" (no marker words), "time", "place", "cause", "object", "way of doing", "direction", "need". There could be several marker words or their groups. In such case, a complex subtype is determined, for example, "place and time", "cause and time" etc.

In the next stage of the analysis, the noun groups are to be allocated in the sentence. A noun group is a set of linked nouns and adjectives, which in aggregate describe a certain entity [16]. The main word in a noun group is a noun. All the external relationships of a noun group are going to this main word. A noun from a noun group is recognized as linked to its main word if there is a case consistency between them and is standing near it or is separated only with adjectives that are coherent with the main word or this word. The adjectives from the noun group could also be linked with adverbs.

Then the building of a syntactic graph is to be performed. The noun groups are nodes of this graph as the separate words do which are not in groups. The essence of the syntactic graph is the representation of the relationships between the words of the sentence. For inflective languages a syntactic graph could be constructed basing on a certain combination of parts of speech in the corresponding word forms. In our model, the types of links between words are based on the parts of a sentence. The name of a link type is related to the dependent part of a sentence to which the link is aimed. There is some connection of these types to the syntactic relationships in a phrase [17]: attributed, objected, circumstantial, appositive and compliant ones.

A brief description of the ontology structure accepted in the developed system

The conception of ontology structure accepted in our system for the moment is following. There are classes of the highest level of the hierarchy, which classify terms in the ontology: “action”, “cause”, “method”, “object”, “place”, “time”. Other variants are also possible according to accepted types of phrases. These classes are devised into more narrow subclasses, but still abstract enough. For example, class “action” could be devised into “active action” and “passive action”. An “active action” means a really performed action, activity (“to organize”, “to develop”, “to begin”, “to perform”). A “passive action” is something expressed by a verb but does not assume a real activity and rather characterize the object's state (“stand”, “exist”, “consist”). The terms of lower hierarchy could be presented by arbitrary nested classes.

The ontology structure used here does not assume individuals. Limitation to classes only for terms description unifies SPARQL queries. Properties could also be present in the ontology. They introduce not hierarchical relationships between concepts. According to the standard of OWL properties have “Domain” and “Range” fields. In the proposed structure of ontology the “Domain” section consists of a set of independent terms, the “Range” section represents a set of terms that are the consequence of the combination of the concepts presented in the “Domain”. Thus, the properties here are peculiar functions where independent variables (factors) are in “Domain” and dependent (responses) in “Range”. For example, in “Domain” there are the following terms: “begin”, “develop”, “conception of informatization”, “Ukraine”, and in “Range” the term is “60-s years of XX century”. It should be noticed that the concept “60-s years of XX century” is a child of the class “Time”. The considered property expresses that the conception of informatization in Ukraine began to be developed in the 60-s years of the XX century. The belonging of terms to the "Domain" is important for the correct interpretation of the word's essence. In this case terms “begin” and “develop” are actions and “Ukraine” is a place.

Building principles and examples of SPARQL queries

As it has been mentioned above, basing on the semantic trees of phrases, which are formed during the analysis of the initial phrase, SPARQL queries are constructed. The structure of the query depends on the type of phrase that is in the base of the semantic tree of the phrase. It is also possible more precise tuning of the variants of the SPARQL queries in the range of each of the types. Let us consider some examples of SPARQL queries depending on the phrase type.

A neutral narration does not assume a certain answer but it is possible to comment it by finding related concepts. In this case, if there are no additional circumstances, the SPARQL query template might look like this:

```
SELECT DISTINCT ?res WHERE {  
  ?y rdfs:domain :inst_name.  
  ?y rdfs:range ?z.  
  ?z rdfs:label ?res.}
```

In the template *inst_name* is a variable that is formed from the main term of the semantic tree of the phrase in a way to fit the classes naming assumed in the ontology, which are formed from a set of the term's words. The query assumes to return “label”, which is a field that contains the terms in a more human understandable manner (the class names are formed by solid join of words stems).

If the semantic tree of the phrase also has additional circumstances, the query template is more complicated:

```
SELECT DISTINCT ?res WHERE {  
  ?y rdfs:domain :inst_name.  
  ?y rdfs:domain :cur_sup_name_1.  
  ...  
  ?y rdfs:domain :cur_sup_name_n.  
  ?y rdfs:range ?z.  
  ?z rdfs:label ?res.}
```

In the template *inst_name*, as it was in the previous case, is a variable formed from the main term of the semantic tree of the phrase; *cur_sup_name_1* ... *cur_sup_name_n* are variables that are forming from the additional circumstances.

The following example illustrates a SPARQL query for the case of the semantic tree of a cause phrase:

```
SELECT DISTINCT ?res WHERE {  
  ?x0 rdfs:subClassOf :cause.  
  ?x_n rdfs:subClassOf ?x0.  
  ...  
  ?last_x rdfs:subClassOf ?x_n.  
  :inst_name rdfs:subClassOf last_x.  
  ?y rdfs:domain :inst_name.  
  ?y rdfs:domain :sup_1.  
  ...  
  ?y rdfs:domain :sup_i.  
  ?y rdfs:range ?z.  
  ?z rdfs:label ?res.}
```

There are the following parameters in the template: *inst_name* is a variable formed from the main term of the semantic tree of the phrase; *sup_1* ... *sup_n* are variables that are forming from the additional circumstances. Such template structure guarantees that the main concept of the semantic tree will be considered in a context of cause regardless of the given concept hierarchy depth.

The next template of SPARQL query is suitable for a semantic tree of a time phrase:

```
SELECT DISTINCT ?res WHERE {
  ?y rdfs:domain :inst_name.
  ?y rdfs:domain :sup_1.
  ...
  ?y rdfs:domain :sup_i.
  ?x rdfs:subClassOf :time.
  ?y rdfs:range ?z.
  ?z rdfs:subClassOf ?x.
  ?z rdfs:label ?res. }
```

In the template there are the following parameters: *inst_name* is a variable formed from the main term of the semantic tree of the phrase; *sup_1* ... *sup_n* are variables that are forming from the additional circumstances. Here for simplification, it is assumed that all the classes denoting time terms are directly inherited from “time” class.

SPARQL queries templates for the semantic trees of phrases of other types are in a hole similar in their structure to the mentioned above ones and so do not given here. On the other hand, it shows sufficient versatility of the proposed model to deal with phrases of different types.

The result of a SPARQL query is a set of concepts that match its selection criteria. If there are no suitable objects in the ontology, then an empty list is returned. In case of obtained data absence, the program executes new SPARQL queries using the reduced noun groups and/or reduced terms list of additional circumstances. The answer, in this case, may appear less relevant, but nevertheless, some answers to the request will be received more probably.

The set of concepts obtained as a result of a query is certainly essential, but this information might be not enough for further machine processing and formation of a natural language phrase for the answer. Therefore, it is to be clarified which class of the higher hierarchy is inherited by each of the found concepts. For this purpose for each of the obtained concepts, a series of SPARQL queries is to be performed, aimed at finding which class is the parent of the class used in the previous iteration. The iterations end when the “Thing” class, which is the root for all the classes, appears as the parent of the class at the current iteration. The template for such request is shown below:

```
SELECT DISTINCT ?res WHERE {
  ?x rdfs:label current_class.
  ?x rdfs:subClassOf ?y.
  ?y rdfs:label ?res. }
```

In the template *current_class* is a value of the “label” parameter for the class from the previous iteration, on the first iteration, it is one of the terms returned by the main query.

Thus, it is possible to get a chain of the hierarchy of higher-level concepts to which belongs the concept returned by the main query. This information is useful for a more informative and correct construction of a natural language answer which is based on the found by the main query concepts. So knowing whether a given concept is an object, action, time, place, etc. it is easier to arrange words in the proper order and, if necessary, supplement them with appropriate prepositions.

Let us consider an example of the process that includes analyzing a phrase, building a request and receiving a response from ontology. The example uses an ontology compiled according to the report of the academician V. S. Mikhalevich on the concept of society informatization. Here is the user's question in Russian: «В чём состоит значение информатизации для человеческого общества?» (eng.: "What is the significance of informatization for the human society?") From this phrase, the system selects one semantic tree: the type of phrase is “question – request for a list of objects”; marker phrase is «в чём состоит» (eng.: “what is”); the main concept is the noun group «значение информатизации» (eng.: “the significance of informatization”); additional circumstances are represented by the noun group «человеческое общество» (eng.: “the human society”). For this case, a SPARQL template is provided, into which lemmatized and stemmed concepts are substituted (marked by italics and underlining):

```
SELECT DISTINCT ?res WHERE {
  ?y rdfs:domain :ЗначенИнформатизац.
  ?y rdfs:domain :ЧеловеческОбществ.
  ?y rdfs:range ?z.
  ?z rdfs:label ?res. }
```

This case is rather simple, there is only one additional circumstance, and concepts are not specifically marked to be descendants of certain classes. The result returned by the query is a set of concepts (terms): «решение научно-технических проблем» (eng.: “solving scientific and technical problems”), «внесение значительного экономического вклада» (eng.: “making a significant economic contribution”), «повышение производительности труда», (eng.: “increasing labor productivity”).

The resulting sets of concepts can be used for the following processing. For example, if the application is a chat-bot, then the concept sets, along with the semantic data and the history of the dialogue, can be used to form a natural

language phrase for the response. However, a detailed description of the SPARQL query results processing is beyond the scope of this paper.

Conclusions and perspectives for the following investigations

A method of analysis is developed for a natural language text in Ukrainian and Russian allowing building semantic trees of phrases, which are convenient for SPARQL queries to ontology forming. The semantic trees of phrases are characterized by the marker words and the expression type. In the initial sentence, several semantic trees could be determined and for each of them, an appropriate SPARQL query can be formed.

Templates of SPARQL queries that make it possible to form the queries to ontology basing on natural language expressions of various types are presented. The type of template is determined by the type and the structure of the given expression, which is represented through the semantic its tree. The template contains inline parameters that are obtained from the corresponding nodes of the semantic tree of the phrase.

The model uses the approach of the noun groups reduction for the situation if the result of a SPARQL query is unsatisfactory (no data obtained). This allows the system to request information on more general concepts, and get some answer, although might be less relevant.

The proposed system could become an element of a user's friendly natural language interface dealing with a knowledge base represented as an ontology or set of ontologies. Also, it may be used for intelligent chat-bots creation.

The future perspectives of the program system development might be an extension and more precise classification of user's phrases, modeling of clarifying questions construction for cases of unsolvable ambiguities, generation of grammatically correct phrases for the answers which form depends on the semantic of the question.

References

1. Gavrilova, T. A. & V. F. Khoroshevsky (2000) *Knowledge Base of Intelligent Systems*. St. Petersburg: Peter.
2. Antoniou, G. (2016) *Semantic Web*. Moscow: DMK-Press.
3. W3C (2013) *SPARQL 1.1 Query Language* [Online] Available from: <https://www.w3.org/TR/sparql11-query/> [Accessed: 11 February 2020].
4. Galitsky, B. (2019) *Developing Enterprise Chatbots. Learning Linguistic Structures*. San Jose: Springer.
5. Popescu, A. M., Etzioni, O. & Kautz, H. A. (2003) Towards a theory of natural language interfaces to databases. *IUI*. p. 149 – 157.
6. Galitsky, B. & Usikov, D. (2015) Programming Spatial Algorithms in Natural Language. *AAAI Workshop Technical Report WS-08-11*. p. 16 – 24.
7. Quirk, C., Mooney, R. & Galley, M. (2015) Language to code: learning semantic parsers for if-this-then-that recipes. *ACL*. p. 878 – 888.
8. Galitsky, B., De La Rosa, J. L. & Dobrocsi, G. (2011) Mapping syntactic to semantic generalizations of linguistic parse trees. *Proceedings of the twenty-fourth international Florida artificial intelligence research society conference*. p. 168 – 173.
9. Li, F. & Jagadish, H. V. (2016) Understanding natural language queries over relational databases. *SIGMOD Record*. 45. p. 6 – 13.
10. Zhong, V., Xiong, G. & Socher, R. (2017) *Seq2SQL: generating structured queries from natural language using reinforcement learning*. [Online] Available from: <https://arxiv.org/pdf/1709.00103.pdf> [Accessed: 11 February 2020].
11. Kupper, D., Strobel, M. & Rosner, D. (1993) Nauda – a cooperative, natural language interface to relational databases. *SIGMOD conference*. p. 529 – 533.
12. Li, Y., Yang, H. & Jagadish, H. V. (2005) Nalix: an interactive natural language interface for querying xml. *SIGMOD conference*. p. 900 – 902.
13. Shaik, S., Kanakam, P., Hussain, S. M., Suryanarayana, D. (2016) Transforming Natural Language Query to SPARQL for Semantic Information Retrieval. *International Journal of Engineering Trends and Technology*. 7. p. 347 – 350.
14. Lashin, V. A. (2010) *Ontologies in computer systems*. Moscow: Scientific World.
15. NLTK Project (2019) *Natural Language Toolkit. NLTK 3.4.5 documentation*. [Online] Available from: <https://www.nltk.org> [Accessed: 11 February 2020].
16. Crystal D. A (2008) *Dictionary of Linguistics and Phonetics* Wiley-Blackwell.
17. Kuryshcheva, M.V. (2014) *Russian language: syntactic analysis of phrases and simple sentences*. Tomsk: Tomsk State Pedagogical University.
18. Shelmanov, A. O. (2015) Ph.D. Tresses: *Study of methods for automatic text analysis and development of an integrated system of semantic-syntactic analysis*. Moscow.

Об авторах:

Литвин Анна Андреевна,

аспирант Института кибернетики им. В. М. Глушкова НАН Украины.

Количество научных публикаций в украинских изданиях – 2.

Количество научных публикаций в зарубежных изданиях – 1.

<http://orcid.org/0000-0002-5648-9074>

Величко Виталий Юрьевич,

кандидат технических наук, доцент,

старший научный сотрудник отдела микропроцессорной техники Института кибернетики им. В.М. Глушкова

НАН Украины; ведущий научный сотрудник отдела создания и использования интеллектуальных сетевых

инструментов Национального центра «Малая академия наук Украины» по совместительству;

Количество научных публикаций в украинских изданиях – 75

Количество научных публикаций в зарубежных изданиях – 27.

H-index: Google Scholar – 11

Scopus – 1,

<http://orcid.org/0000-0002-7155-9202>.

Каверинский Владислав Владимирович,

кандидат технических наук,

старший научный сотрудник отдела износостойких и коррозионностойких порошковых конструкционных материалов Института проблем материаловедения им. И. Н. Францевича НАН Украины.

Количество научных публикаций в украинских изданиях – 81.

Количество научных публикаций в зарубежных изданиях – 18.

H-index: Google Scholar – 4

Scopus – 2,

<http://orcid.org/0000-0002-6940-579X>

Место работы авторов:

Институт кибернетики им. В. М. Глушкова НАН Украины.

03187, Киев-187, проспект Академика Глушкова, 40.

Литвин Анна Андреевна, Тел.: (097) 570-99-84, E-mail: litvin_anya@ukr.net

Величко Виталий Юрьевич, Тел.: (096) 139-96-28, E-mail: aduisukr@gmail.com

Института проблем материаловедения им. И. Н. Францевича НАН Украины.

03142, Киев, ул. Кржижановского, 3.

Каверинский Владислав Владимирович, Тел.: (050) 212-17-24, E-mail: insamhlaithe@gmail.com