Semantic Analysis and Natural Language Text Search for Internet Portal

Tetiana Kovaliuk[1'[0000-0002-1383-1589]', Nataliya Kobets[2'[0000-0003-4266-9741]'

[1]National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute",
37, Prospekt Peremohy, Kyiv 03056, Ukraine
tetyana.kovalyuk@gmail.com,
[2]Borys Grinchenko Kyiv University, 18/2 Bulvarno-Kudriavska Str, Kyiv, 04053, Ukraine
nmkobets@gmail.com

Abstract. The article is devoted to solving the set of problems related to natural language texts semantic analysis. The following problems are addressed: automation of generating metadata files describing the semantic representation of a web page; semantic network construction for a given set of texts; semantic search execution for a given set of texts using metadata files; and semantic network export to RDF format. The algorithms for knowledge extraction from text, semantic network construction and query execution on a given semantic network are described. The lexico-syntactic patterns method was used as a basis to approach these problems. A specification for describing lexico-syntactic patterns has been developed and a pattern interpreter based on the morphological dictionary of the Ukrainian language has been created as a part of the software implementation of the method. Experimental studies have been carried out for the «classification of living organisms» subject environment set of patterns. Modified Boyer–Moore–Horspool algorithm was used to address the problem of interpreting.

Keywords: metadata file, semantic network, semantic search, lexico-syntactic patterns.

1 Introduction

International Data Corporation predicts that the summation of all data, whether it is created, captured, or replicated - called the Global Datasphere – is experiencing tremendous growth. Global Datasphere will grow from 33 Zettabytes (ZB) in 2018 to 175 ZB by 2025. [1]. A considerable amount of information exists in the form of natural language texts. The problem of developing and applying new, more progressive approaches to the presentation and analysis of information on the Internet, including natural language texts, is becoming more and more acute. In this context, the problem of information search arises [2]. One of possible solutions to this problem is conversion of the World Wide Web to a semantic representation of data that is mapping each web resource to a metadata file that contains the semantics of the

resource and is suitable for machine analysis. Thus, the problem of automating the creation of metadata files is relevant.

The purpose of this study is to determine the possibility of reducing the metadata files creation workload for web pages, to reduce the time of search for relevant query results using metadata and to improve the quality of the results for search queries.

The objective is to perform semantical analysis of natural language texts in order to fulfill queries formulated in natural language. This objective is complex and can be divided into the following sub-objectives:

- to transform the query into a semantic representation;
- to perform natural language texts analysis in order to extract knowledge and present it in the form of a semantic network [3];
- query execution using logical deduction on a set of predicates of a semantic network;
- to transform the results of a query from a machine representation into a natural language form.

The problem of query transformation into a semantic representation, in turn, decomposes into the following sub-objectives:

- to determine the context of a query, i.e., subject domain [4];
- to identify the query semantics in the context of a specific subject domain [5].

The problem of natural language texts analysis in order to extract knowledge and present it in the form of a semantic network can be broken down into the following sub-objectives:

- to classify available texts by certain subject environments they belong to;
- to perform semantic analysis of texts that correspond to the subject domain of the query. In fact, the result of this analysis is knowledge of the subject environment, represented in the form of relations between objects
- to construct a semantic network that contains the knowledge gained in the previous step, as well as the knowledge of ontology relevant to the subject domain.

As part of this study, the following problems were solved:

- transformation of the query into a semantic representation;
- natural language texts analysis to extract knowledge and present it in the form of a semantic network;
- query execution using logical deduction on a set of predicates of a semantic network.

A semantic network is an informational model of a subject domain that has the form of a directed graph whose vertices correspond to the entities of the subject domain, and edges define the relations between them. Entities can represent concepts, events, properties, processes. In a semantic network, the concepts of the knowledge base are vertices and edges (directed) define the relations between them. Thus, the semantic

network reflects the semantics of the subject domain in the form of concepts and relations between them [6] [7].

2 Mathematical formulation of the problem

Suppose there is a set of natural language texts that describes M subject domains. For each of M subject domains, the system retains its ontology $O =< P, C, R >$, where P is a set of predicates, C is a set of concepts that form the basis of the subject domain, R is a finite set of functions that are defined on concepts and predicates of ontology. The basis is a set of predicates and concepts that are sufficient to describe at least 80% of the texts of a given subject domain.

An input to the system is given in the form of a natural language text. As a result of the analysis of incoming texts, the system returns the result of the query execution in the natural language.

The problem of semantic analysis and natural language text searching (queries are formulated in the natural language) breaks down into following sub-tasks:

1. Sub-task of the query context determination is about finding the correspondence between the query and some ontology $O_k =< P, C, R >, k \in [1, M]$, where k is the index of a subject domain, which corresponds to the query of the subject domain. Task and domain ontologies all relate to the context of a specific domain (e.g., zoology, biology) or task (e.g., accounting). The ontology "classification of living organisms" is used.
2. Sub-task of extracting semantics of the query in the context of a specific subject domain becomes a semantic analysis of a query, which will result in a set of triplets $\{S, P, O\}$, where S is subject, P is predicate, O is object. Predicate (property) is a binary relation between subject and object (fig.1).

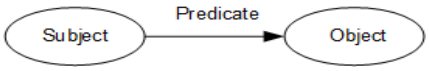

Fig. 1. Graph of triplet.

3. Sub-task of classification of available texts by certain subject environments they belong to comes down to splitting the original texts set T into M subsets $\{T_1, T_2, ..T_M\}$, each of which corresponds to a certain subject domain and ontology $O_k =< P, C >, k \in [1, M]$.
4. Sub-task of semantic analysis of texts that correspond to the subject domain of the query is about extracting the knowledge from a certain subset of texts T_k that correspond to the subject domain of the query; the knowledge is represented in the form of a set of triplets $\{S, P, O\}$, where S is the subject, P is the predicate, O is the object.

5. To construct a semantic network containing the knowledge obtained in the previous step as well as the knowledge of the ontology corresponding to the subject domain it is required to augment the resulting set of triplets $\{S, P, O\}$ with information about the predicates and concepts $\{P, C\}$ that are the basis for the given subject domain. The resulting triplet set can thus be represented as a directed graph, the vertices of which correspond to the objects and concepts of the subject domain, and edges are relations between them.
6. Sub-task of query execution using logical deduction on a set of predicates of a semantic network involves the application of methods for the automatic proofing of theorems on the triplet set $\{S, P, O\}$ that consists of the relations of the semantic network and the query. With logical deduction, a set of triplets is obtained, which is the result of the query execution.

3 Method for solving the problem of knowledge extraction from text and representing knowledge as a semantic network

Method selection for solving the problem of semantic analysis and search is based on the criteria of relevance of query results and performance requirements.

One of the most effective approaches to solving the problem of semantic analysis of texts, which allows to set up a system for specific subject environments, is the lexico-syntactic patterns method [8] [9] [10][11].

The lexico-syntactic pattern is a structural model of the grammatical construction. The pattern specifies its lexical composition and surface syntactic properties and thus can be used to detect it in the text and to further extract it. The pattern is built as a sequence of elements describing the corresponding fragments of the grammatical construction in the order in which they are found in this grammatical construction.

The method of lexico-syntactic patterns assumes that lexical relations can be described using a pattern hierarchy that consists mainly of indicators of the part of the speech and the group symbols.

The advantages of the lexico-syntactic patterns method are the simplicity of implementation and the high relevance of the results if there is a sufficiently complete set of lexico-syntactic patterns covering the basis of the researched subject domain. The disadvantage of the method is the need to create lexico-syntactic templates for each subject environment.

Therefore, lexico-syntactic patterns allow to design a semantic model that corresponds to the text to which they are applied.

The lexico-syntactic patterns method works with the following inputs:

- a set of lexico-syntactic patterns that corresponds to a given subject environment;
- an ontology containing the basic concepts and predicates for a given subject environment;
- a text (or set of texts) for semantic analysis.

To work correctly, the method must first classify analyzed text or query by belonging to a particular subject environment. Classification can be done using the Bayes algorithm or any other method.

3.1 Description of the structure of lexico-syntactic patterns

A lexico-syntactic pattern is a sequence of string-elements and word-elements [12].

A string-element allows you to write the desired string of characters in the pattern, in particular, a specific word form, a punctuation mark, or a legend that occurs in a formalized grammatical construction, for example, «equals», «set», «+», and so on.

A word-element corresponds to a single word of the grammatical construction. In general, such properties are specified for a word-element:

- part of speech (first letters of the corresponding names are used - table 1);
- specific lexeme that defines a set of all word forms (identifiable by name);
- values of the morphological parameters of a word that narrow the set of possible word forms, for example, c – case, n – number, g – gender, t – tense, p – person and so on.

Table 1. Part of speech (legend).

Part of speech	Legend
Word	W
Noun	N
Adjective	A
Verb	V
Pronoun	Pn
Adverb	Av
Preposition	Pr
Interjection	Int
Particle	Pt
Numeral	Num

For example, the word-element $V<to\ understand, t=present, p=3>$ describes the verb «to understand» in the present tense and in the third person, that is, it defines its word form «understand» or «understands».

When creating a word-element, a specific lexeme and the value of the morphological parameters may not be specified, which allows specifying any word form for the given lexeme (for example $N<file>$) or the arbitrary word of a certain part of speech with the required grammatical characteristics. For example, a lexeme that defines an adjective in the form of a singular instrumental (ablative) case can be written $A<;\ c=instrumental,\ n=singular>$. In general, the pattern may include both several words from different parts of speech and several different words from one part of speech. Numerical indices are used to distinguish them. For instance,

pattern $NN = N1N2 <; c = genitive >$ includes two different nouns $N1$ and $N2$, the second one being in the genitive case.

Concord rules point to the grammatical concord of the individual elements of the formalized grammatical construction and refer to the whole pattern. They are defined after describing all elements of the pattern in the form of equality of values of the concordant morphological parameters. For instance, the pattern $PV = PnV < Pn.n = V.n, Pn.g = V.g >$ describes concordant pair (pronoun and verb) in the number n and gender g : «it looks, they agree, she thinks» and so on.

Repetition of the elements can be specified in the pattern. For example, the pattern $\{N <;\ c = genitive >\}$ defines a chain of nouns going in a row in the genitive case. If there are known limitations on the number of identical elements, then they can be specified in the pattern. The record $\{A\} < 1,3 > N$ defines a sequence of one, two, or three adjectives and a noun.

Also, a pattern may include non-mandatory elements (in square brackets): for example, the [«not»] element specifies the non-binding nature of the «not» particle of the language expression. Acceptable recording of alternatives of a certain grammatical construction is one, which uses the symbol | (logical operation «or»). For instance, the pattern $NP = N \mid Pn$ describes the alternative between the noun N and the pronoun P_n.

The full specification of the lexical-syntactical patterns is given here [13]:

Thus, when creating a pattern of a complex grammatical construction it makes sense to highlight its constituent parts and to describe them one by one in the form patterns.

3.2 Boyer–Moore–Horspool based algorithm for interpreting lexical-syntactic patterns

The work of the interpreter is to compare lexical-syntactic patterns for each predicate of the subject environment with the analyzed text. Usually, the subject environment is described by a set of predicates, for each of them a set of patterns are created. Since the speed of interpretation plays a critical role, it is desirable to apply the most efficient interpretation algorithm. One of the best-performing algorithms is the Boyer-Moore- Horspool (BMH) algorithm [14] [15], which is the reason why it was used in this work.

BMH is a modified version of original Boyer-Moore algorithm. It performs fast matching compare to other algorithms regardless of the pattern size that process it. Algorithm BMH is as good as the original Boyer-Moore algorithm. Moreover, the same results show that for almost all pattern lengths this algorithm is better than algorithms that use a hardware instruction to find the occurrence of a designated character. This algorithm works fast in situations where the pattern is much shorter than the processed text, or when searching in multiple documents. Usually, the longer the pattern, the faster the algorithm works [16].

Input data of the algorithm for interpreting lexico-syntactic patterns consists of:

- the array L of lexico-syntactic patterns $l_i, i = \overline{1.m}$ for the given ontology $O_k =< P_k, C_k, R_k >$, where $l =< l_0, l_1, ..., l_m >, l_i, i = \overline{1,m}$ – the element of the pattern, m – number of elements in the pattern; $l_i \in K_1 \cup K_2$, K_1 – string-elements class, K_2 – word-elements class according to [12];
- string-element contains the constant string str: $l_i = \{str\}, \forall l_i \in K_1$ only;
- word-element can contain the constant string str, as well as data on the parameters (grammatical categories) of the word: part of speech, gender, case, tense, grammatical number;
- natural language text T.

The result of the algorithm is the set of triplets $\{S, P, O\}$, where S is the subject, P is the predicate, O is the object.

The following is a description of the algorithm step-by-step:

Step 0. Sort elements of each pattern by belonging to K_1 and K_2 classes. As a result, a sorted set of patterns L_{sort} will be received for which the following statement holds true: $\forall l_i, l_j \in L, \forall i,j \,|\, \text{if } i > j \text{ then } l_i \in K_1, l_j \in K_2$.

Step 1. Sort string-elements by the length in ascending order. As a result, a sorted set L_{sort} of patterns will be received for which the following statement holds true: $\forall l_i, l_j \in L, \forall i, j \,|\, \text{if } l_i, l_j \in K_1, i > j$ then the length of the string l_i is greater than the length of the string l_j, also $l_i, l_j \in L, \forall i,j \,|\, \text{if } i > j \text{ then } l_i \in K_1, l_j \in K_2$.

Step 2. Sort word-elements by the level of concretization. As a result, a sorted set L_{sort} of patterns will be received for which the following statements hold true:

$$\forall l_i \in L, \forall l_j \in L, \forall i, j \text{ if } l_i, l_j \in K_2, i > j \text{ then } |l_i| > |l_j| \quad (1)$$

$$\forall l_i \in L, \forall l_j \in L, \forall i, j \text{ if } l_i, l_j \in K_1, i > j \text{ then } |l_i| > |l_j| \quad (2)$$

$$\forall l_i \in L, \forall l_j \in L, \forall i, j \text{ if } i > j \text{ then } l_i \in K_1, l_j \in K_2 \quad (3)$$

where $|l_i|$ – length of string l_i, $|l_j|$ – length of string l_j.

Step 3. Pick one element l from unprocessed elements in the set of lexico-syntactic patterns L. If all elements of L are processed, then go to step 4. Otherwise, go to step 3.1.

Step 3.1. Select the next unprocessed string-element l_i^{cur} of the pattern l^{cur}, then go to step 3.1.1. If all string-elements are processed, then go to step 3.2.

Step 3.1.1. For each character $str[i]$ except the last one ($i \neq n$, where n is the length of the string str) in string $str \in l_i^{cur}$ define the value $val[i]$, which is equal to the maximum shift of the character relative to the beginning of the word:

$$val[i] = \max_{j / str[j]=str[i]} \{j\} \quad , \forall i \in [0; n-1] \tag{4}$$

$$val[n-1] = n \tag{5}$$

As a result of this step, the mapping «character – the minimum shift relative to the end of the string» will be received.

Step 3.1.2 Match the beginning of the text T and the beginning of the string-element l^{cur} : $shiftText = 0$ and $shiftStr = 0$. Go to step 3.1.3.

Step **3.1.3** If the end of text T is reached, then go to step 3.1. Examine the last character $str[n-1-shiftStr]$ of the string-element. If the given character matches the corresponding character of the text $str[n-1-shiftStr] = T[shift]$ then go to step 3.1.4, otherwise, go to step 3.1.5.

Step 3.1.4 If the current character is the first in the word, that is $n-1-shiftStr = 0$, then go to step 3.1.

Check the next character from the end of the string-element l^{cur} : $shiftStr = shiftStr + 1$. If the given character matches the corresponding character of the text $str[n-1-shiftStr] = T[shift]$, then go to step 3.1.4, otherwise, go to step 3.1.5.

Step 3.1.5 Execute text shift T by number $\Delta shift$, which corresponds to the current character $T[shift]$ in the mapping $shift = shift + \Delta shift$. If the character is not in the table, then the shift is equal to the length of the word str. Make shift $shiftStr$ within a single word str equal to zero: $shiftStr = 0$. Go to step 3.1.3.

Step 3.2. Select the next unprocessed word-element l_i^{cur} of the pattern l^{cur}, then go to step 3.2.1. If all word-elements are processed, then go to step 3.3.

Step 3.2.1. Check the match of the morphological form of the word-element l_i^{cur} and the word from the text, which position corresponds to the position l_i^{cur} in the lexico-syntactic pattern. If the morphological form matches all parameters (part of speech, gender, case, tense, grammatical number), then go to step 3.2. Otherwise, go to step 3.

Step 3.3. Add the triplet $\{S, P, O\}$ corresponding to the pattern l^{cur} to the semantic network.

Step 4. End the algorithm.

4 Software Implementation and Example

The software has a three-tier architecture: it consists of the client application, server application, and database server. Language C# and the .NET platform selected for implementation due to advantages such as automatic garbage collection, the built-in

language of LINQ queries, the availability of ADO.NET classes for access to database and a number of other benefits.

The client application consists of the following modules:

- an access module to work with the server-side application;
- semantic network graphical display module;
- user interface module.

The access module to work with the server-side application is designed to communicate with web-service, which implements business logic.

The user interface module is responsible for creating a GUI that allows the user to conveniently enter the input data and view the results of its processing.

The semantic network graphical display module is responsible for creating an image of a directed graph corresponding to the constructed semantic network.

The server application consists of the following modules:

- a semantic network construction module;
- a module for exporting the semantic network from logical to RDF format [17];
- a query analysis module;
- a module to perform logical deduction on the semantic network;
- an access module to work with the database.

The semantic network construction module implements the method of lexico-syntactic patterns in order to extract knowledge from the text.

The module for exporting the semantic network from logical to RDF format allows to retain knowledge that was extracted from the text in the format suitable for machine processing.

The query analysis module determines the content of the query.

The module to perform logical deduction on a semantic network executes a query and present results in the form of a triplet set «object-predicate-subject».

The access module to work with a database is designed to provide access to the morphological dictionary and other data stored in a database for the application server.

For semantic analysis, you must specify a * .txt, * .doc or *.html file or a web resource URL. The program builds the semantic network in the process of text analysis To represent the knowledge obtained from the text in a machine-friendly format the RDF format is used (fig. 2).

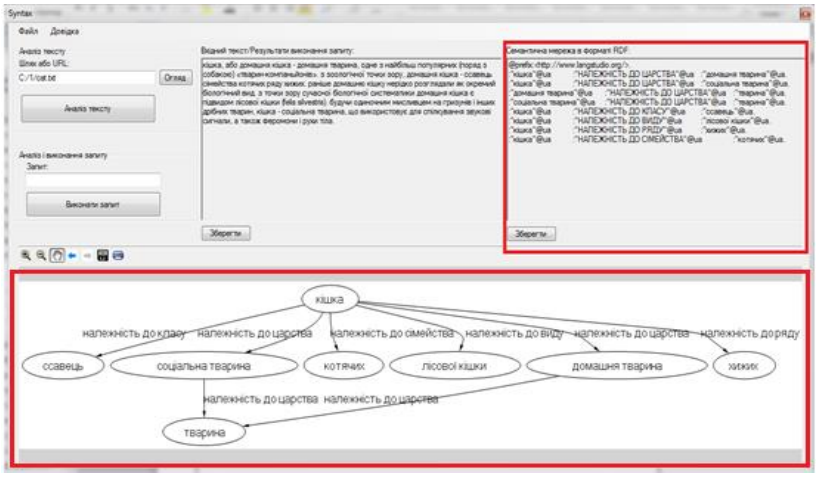

Fig. 2. Presentation of knowledge received from text in RDF format and image of semantic network.

5 Conclusion

The paper deals with the problems of extracting knowledge from texts, presenting them in the form of a semantic network, and executing queries on the constructed network. The main approach to solving these problems was the lexico-syntactic patterns method. A specification to describe lexico-syntactic patterns has been developed, a pattern interpreter has been created based on the morphological dictionary of the Ukrainian language and the «classification of living organisms» subject environment set of patterns has been collected as a part of the software implementation of the method. Modified Boyer–Moore–Horspool algorithm was used to solve the problem of interpreting lexico-syntactic patterns.

Future work will be concentrated on creating tools for simplifying the process of building new lexico-syntactic patterns, adding new lexico-syntactic patterns to the database, improving the query execution module, and creating a module for text classification by belonging to the subject environment

References

1. Data Age 2025. The Digitization of the World from Edge to Core, https://www.seagate.com/files/www-content/our-story/trends/files/idc-seagate-dataage-whitepaper.pdf, last access 2019/02/10.
2. Manning, C. D., Raghavan, P., Schutze, H.: Information Retrieving. Cambridge: Cambridge University Press (2009)
3. Clark, A., Fox, C., Lappin, S.: Computational Linguistics and Natural Language Processing. Wiley-Blackwell Publishing (2010)

4. Ijntema, W., Sangers, J., Hogenboom F., Frasincar, F.: A Lexico-Semantic Pattern Language for Learning Ontology Instances from Text. In: Web Semantics: Science, Services and Agents on the World Wide Web, Vol. 15, pp. 37-50 (2012)
5. Mark Johnson. Natural Language Processing and Computational Linguistics: from Theory to Application, http://web.science.mq.edu.au/~mjohnson/papers/CLandTopicModels.pdf, last accessed 2019/04/12.
6. Hebeler, J., Fisher, M., Blace, R., Perez-Lopez, A.: Semantic Web Programming. Indianapolis: Wiley Publishing (2009)
7. Pollock, J.T.: Semantic Web for Dummies. Indianapolis: Wiley Publishing (2009)
8. Lexico-Syntactic Pattern Language: language description LSPL, http://www.lspl.ru/, last access 2019/02/10.
9. Zagorul'ko, Yu. A., Sidorova, E.A.: Extraction system subject terminology from text based on lexical and syntactic patterns. In: Proc. XIII International Conference on Control and Modeling Problems in Complex Systems. Samara, pp. 506–511 (2011)
10. Klaussner, C., Zhekova, D.: Lexico-Syntactic Patterns for Automatic Ontology Building. In: Proceedings of the Student Research Workshop associated with RANLP 2011, Hissar, Bulgaria, pp. 109–114 (2011)
11. Mochalova, A.V.: Algorithm for semantic text analysis by means of basic semantic templates with deletion. In: Scientific and Technical Journal of Information Technologies, Mechanics and Optics. No. 5 (93), pp. 126-132 (2014)
12. Bolshakova, E.I., Baeva, N.V., Bordachenkova, E.A., Vasilyeva, N.E., Morozov, S.S.: Lexico-syntactic patterns in the tasks of automatic text processing. In: Proceedings Int. Conference “Dialogue 2007”, pp.70-75 (2007)
13. Language description LSPL (1.0.1), http://www.lspl.ru/articles/LSPL_Refguide_13.pdf, last access 2019/02/10.
14. Horspool, R.N.: Practical Fast Searching in Strings. Software-Practice and Experience, vol. 10, pp. 501-506 (1980)
15. The Boyer-Moore-Horspool Algorithm, http://www.mathcs.emory.edu/~cheung/Courses/323/Syllabus/Text/Matching-Boyer-Moore2.html, last access 2019/04/14.
16. Hasan, A.A., Nur Aini Abdul Rashid, Muhannad A. Abu-Hashem, Atheer A. Abdulrazzaq: Multi-Pattern Boyer-Moore-Horspool Algorithm based Hashing Function for Intrusion Detection System. In: Lecture Notes on Information Theory, Vol. 1, No. 2 (2013)
17. Description of the data presentation model RDF, http://www.w3.org/RDF/, last access 2019/04/14
18. Vavilenkova, A.I.: Software for detection of text documents identical in content. In: Visnyk of Chernihiv State Technological University, No. 2 (65), pp 125-132 (2013)