ISSN 1870-4069

Extraction of Semantic Trees from a Text while Constructing Domain Ontology

Nadezhda Yarushkina, Aleksey Filippov, Vadim Moshkin

Ulyanovsk State Technical University, Ulyanovsk, Russia
{jng, al.filippov, v.moshkin}@ulstu.ru

Abstract. This article describes the method of building a domain ontology based on the linguistic analysis of content of text resources. Also an example of the proposed approach and the architecture of our pipeline presents. Representation of the problem area (PrA) in the form of a domain ontology is often used in the process of development of intelligent software systems and used as a knowledge base. The process of building an ontology is complex and requires an expert in the PrA. A large number of researchers are working to solve this problem. The basis of our approach is the use of a pipeline of different linguistic methods of text analysis. The set of rules developed by us is used to build an ontology based on the content analysis of a text resource.

Keywords: domain ontology, semantic analysis, linguistics, text resources.

1 Introduction

Currently, methods of artificial intelligence are used to solve various problems in the field of business process automation. The use of methods of artificial intelligence allows intelligent systems to solve intellectual tasks at a level close to a human. Intelligent systems must have knowledge about the PrA to successfully solve the intellectual tasks. The methods of knowledge engineering allow to describe the features of the PrA in the form of a domain ontology [1, 2, 3, 4, 5, 6].

At present, ontologies are formed by experts in the problem area (PrA). The expert must have skills in the field of ontology engineering and have a good understanding of the specifics of a particular PrA. Building an ontology is a long and complex process.

The main drawback of domain ontologies is the need for their development and updating due to PrA change. Knowledge extraction is carried out to extend the ontology. Knowledge extraction is carried out using semi-automatic methods for transforming unstructured, semi-structured and structured data into conceptual structures.

Now there are several directions for building the ontology:

1. extraction of knowledge from Internet resources (in particular, wiki-resources);
2. analysis of dictionaries and thesauri;
3. merging of different ontological structures;
4. extraction of terminology in the process of text processing using statistical and linguistic methods.

Thus, the task of automatically building ontologies based on the analysis of the contents of text resources is currently relevant.

A large number of researches are devoted to the automatic building of the domain ontology on the basis of the analysis of the content of wiki-resources. Wiki-resource -- a website whose structure and content can be modified by using a special markup language. User do not need additional tools and IT skills to work with wiki-resources. So different wiki-resources may be used as data sources for the building of ontologies as they contain knowledge of various PrAs and freely available for use.

There are various approaches to the automatic generation of ontologies based on the analysis of the contents of wiki-resources:

1. Formation of classes and relations of ontology on the basis of analysis of the structure of wiki-resources [7,8,9,10,11].
2. Formation of objects and relations of ontology on the basis of analysis of the structure of wiki-resources [7,12,13,14,15].
3. Formation of an ontology in the process of combining several ontologies [16,17,18,19,20].

For example, in the YAGO project for automatic building of the domain ontology, data from Wikipedia and data from the semantic WordNet network were used. The ontology was built on the basis of a hierarchy of Wikipedia pages and information from info-boxes, and then expanded based on WordNet data. As you can see, the contents of the pages of wiki-resources are almost not taken into account, instead, various widely available thesauri are used.

We believe that the analysis of the content of the wiki-resources will increase the completeness of the description of the PrA in the form of a domain ontology. Also, an ontology can be built on the basis of an analysis of the contents of a set of text documents. The idea of our approach is to use the existing methods of linguistic analysis to construct a syntactic tree of sentence. Further, using a set of rules, you can translate a syntax tree into a semantic tree. Semantic representation of the text on natural language (NL) is the most complete of those that can be achieved only by linguistic methods. The domain ontology can be built from the semantic trees extracted from content of text resources. It is necessary to develop a method of translating a syntactic tree into a semantic tree.

2 A Method of Translating a Syntactic Tree into a Semantic Tree

It is necessary to determine the syntactic structure of the sentence on NL for constructing the semantic tree. There are several parsing tools of texts in Russian, for example [21,22,23,**¡Error! No se encuentra el origen de la referencia.**]:

1. Lingo-Master;
2. Treeton;
3. DictaScopeSyntax;
4. ETAP-3;
5. ABBYY Compreno;

6. Tomita-parser;
7. AOT, etc.

In our work, for constructing a syntactic tree the results of the AOT project were used. Consider the application of the algorithm of translating a syntactic tree into a semantic tree using the example of test sentence in Russian: "*Онтология в информатике - это попытка всеобъемлющей и подробной формализации некоторой области знаний с помощью концептуальной схемы* ".

The translation of test sentence into English is used to improve the perception of the algorithm: "Ontology in informatics is an attempt at comprehensive and detailed formalization of a certain field of knowledge with the help of a conceptual scheme".

The resulting syntactic tree of test sentence is shown in the Figure 1.

Formally the function of translating a syntactic tree into a semantic tree:

$$F^{Sem}:\left\{N_{li}^{Synt},P_j\right\}\rightarrow\left\{N^{Sem},R^{Sem}\right\}, \quad (1)$$

where N_{li}^{Synt} – i-th node of l - th level of the syntactic tree. For example, the first node of the first level is the node "ontology", the second - "pg", the third - "is", etc. for the parse syntactic in Figure 1. The node of the syntactic tree can be a member of the sentence, for example, the node "ontology", or also can be a syntactic label that defines the constituent members of the sentence, for example, "pg" (the prepositional group); P_j – j-th rule for translating the nodes of the syntactic tree. The nodes of the syntactic tree will be translated into nodes and relations of the semantic tree.

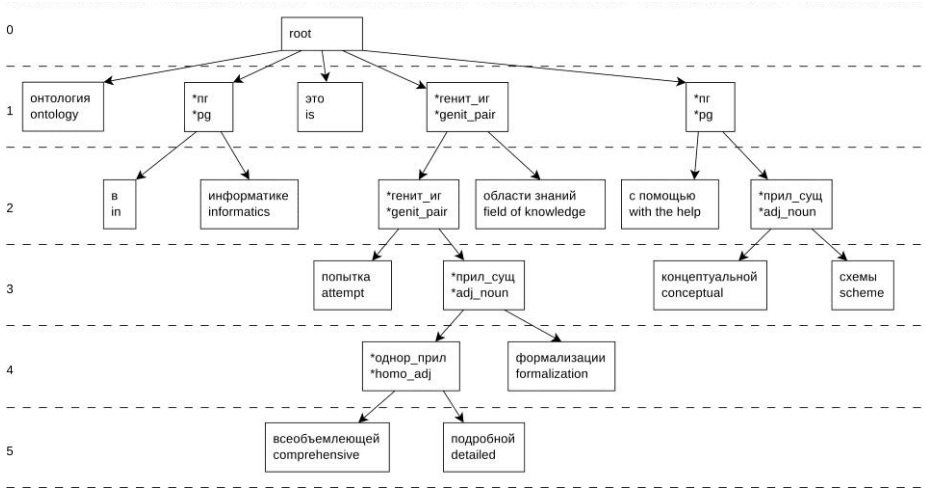

Fig. 1. Example of a syntactic tree of test sentence.

The rule is a collection of several words (units) united according to the principle of semantic-grammatical-phonetic compatibility. Formally rule:

$$\{N_1^{Synt},N_2^{Synt},\ldots,N_k^{Synt}\}\rightarrow\left\{N^{Sem},R^{Sem}\right\},k=\overline{1,K}, \quad (2)$$

where $N_1^{Synt}, N_2^{Synt}, \ldots, N_k^{Synt}$ is the set of units of the rule corresponding to the set of nodes of the syntactic tree. The rule only works if all the units match. Examples of rules and the results of their use are presented in Table 1. *K* is number of units in the rule; $\left\{ N^{Sem}, R^{Sem} \right\}$ is set of nodes N^{Sem} and relations R^{Sem} of the semantic tree, obtained as a result of translation of the syntactic tree into a semantic tree.

Table 1. Examples of rules for translating nodes of syntactic tree into nodes of a semantic tree and the results of their application.

Initial data	Rule	Result
attempt-***genit_pair**-formalization	node1- ***genit_pair** -node2 → node1-associateWith-node2	*attempt*-associateWith-*formalization*
in-***pg**-*informatics*	node1-***pg** -node2 → prevNode-***dependsOn***(node)-node2	lastNode-***dependsOn***-*informatics*
is	**is** → prevNode-nextNode	lastNode-isA-nextNode
conceptual-***adj_noun**-*scheme*	node1-***adj_noun**-node2 → node2-hasAttribute-node1	*scheme*-hasAttribute-*conceptual*
comprehensive -***homo_adj**-*formalization detailed*-***homo_adj**-*formalization*	node1-***homo_adj**--node2 → node2-hasAttribute-node1	*formalization* -hasAttribute-*comprehensive formalization*-hasAttribute- *detailed*

$$R^{Sem} = \left\{ R_{isA}^{Sem}, R_{partOf}^{Sem}, R_{associateWith}^{Sem}, R_{dependsOn}^{Sem}, R_{hasAttribute}^{Sem} \right\}, \quad (3)$$

where R_{isA}^{Sem} – set of transitive relations of hyponymy;

R_{partOf}^{Sem} – set of transitive relations «part/whole»;

$R_{associateWith}^{Sem}$ – set of symmetrical relations of association

$R_{dependsOn}^{Sem}$ – set of asymmetric relations of associative dependence;

$R_{hasAttribute}^{Sem}$ – set of asymmetric relations describing the attributes of nodes.

3 The Algorithm of Translating a Syntactic Tree into a Semantic Tree

The algorithm of translating a syntactic tree into a semantic tree consists of the following steps:

1. Go to the first level of the syntactic tree.
2. Select the next node of the current tree level. If there are no unprocessed nodes, go to step 12.
3. If the node is marked as processed, go to step 2.
4. If the node is not a syntax label (not starts with "*"), go to step 10.
5. If the node is a syntax label (starts with "*") and does not have child elements, go to step 10.

6. If the node is a syntax label (starts with "*") and all its child nodes are not syntax labels, go to step 10.
7. If there is a temporary parent node, then replace it, otherwise create a temporary node.
8. If there is no connection between the nodes, create a temporary relationship between them and go to step 2.
9. If both nodes are not temporary and there is no connection between them, create an "associateWith" relationship between them and go to step 2.
10. Apply the rule for translation.
11. Mark the nodes as processed and go to step 2.
12. Go to the next level of the syntactic tree, and then go to step 2.

4 Example of the Algorithm of Translating a Syntactic Tree into a Semantic Tree

Let's consider an example of translating the syntactic tree of test sentence presented above into a semantic tree. The following nodes of syntactic tree (syntactic units) were identified in the first level of the syntactic tree of the test sentence (see Figure 1):

- *ontology*;
- ***pg** (*informatics*);
- *is*;
- ***genit_pair**(***genit_pair**(*attempt*, ***adj_noun**(***homo_adj**(*comprehensive*, *detailed*), *formalization*)), *field of knowledge*);
- ***pg**(*with the help*, ***adj_noun**(*conceptual*, *scheme*)).

Figure 2 shows the semantic tree of test sentence at the beginning of the algorithm.

Figure 3 shows the semantic tree of test sentence at the first iteration of the algorithm.

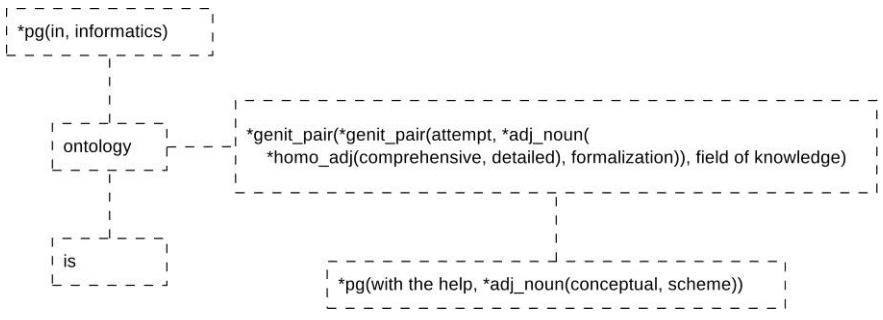

Fig. 2. Example of a semantic tree of test sentence at the beginning of the algorithm.

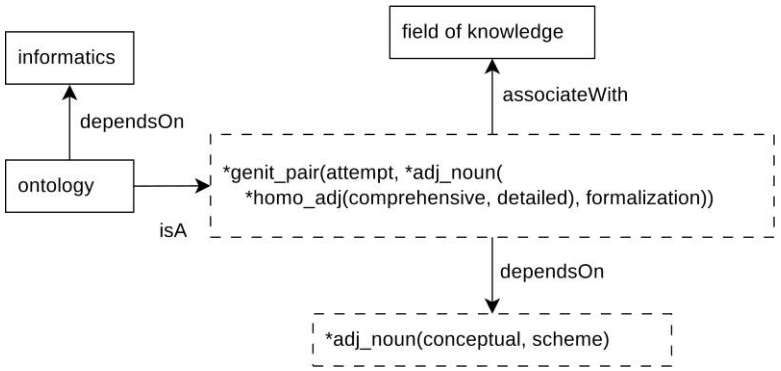

Fig. 3. Example of a semantic tree of test sentence at the first iteration of the algorithm.

As you can see from Figure 3, all syntactic units of the first level of the syntactic tree of the test sentence were processed. After applying the translation rules:

- the syntactic unit "ontology" was included in semantic tree;
- from the syntactic unit "\is" the relation "isA" was formed between the node "ontology" and the temporary node "***genit_pair**(...)";
- from the syntactic unit "***pg**(*in, informatics*)" the node "informatics" and relation "dependsOn" between the nodes "informatics" and "ontology" were formed;
- from the syntactic unit "***genit_pair**(***genit_pair**(...)), *field of knowledge*)" the temporary node "***genit_pair**(...))" and the node "field of knowledge" were formed that are connected by the relation "associateWith";
- from the syntactic unit "***pg**(*with the help, ...*)" the temporary node "***adj_noun**(*conceptual, scheme*)" and relation "dependsOn" between that node and the temporary node "***genit_pair**(...))" were formed.

All syntactic units of the first level and all syntactic units of the second level that are related to the syntactic units of the first level were marked as processed in the syntactic tree of test sentence.

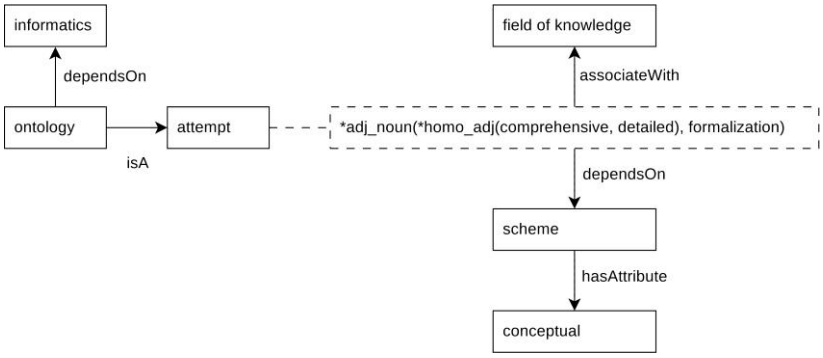

Fig. 4. Example of a semantic tree of test sentence at the second iteration of the algorithm.

Figure 4 shows the semantic tree of test sentence at the second iteration of the algorithm.

As you can see from Figure 4, all syntactic units of the second level of the syntactic tree of the test sentence that not marked as processed were processed. After applying the translation rules:

- from the syntactic unit "***genit_pair** (*attempt*, ***adj_noun**(***homo_adj** (*comprehensive*, *detailed*), *formalization*))" the node "attempt" and temporary node "***adj_noun**(...)" were formed that are connected by relation "associateWith". In the genitive pair, the second node is the main node, so the existing relationships refers to the second node;
- from the syntactic unit "***adj_noun**(*conceptual*, *scheme*)" nodes "conceptual" and "scheme" and relation "hasAttribute" between them were formed.

All syntactic units of the second level and all syntactic units of the third level that are related to the syntactic units of the second level were marked as processed in syntactic tree of test sentence.

Figure 5 shows the semantic tree of test sentence at the third iteration of the algorithm.

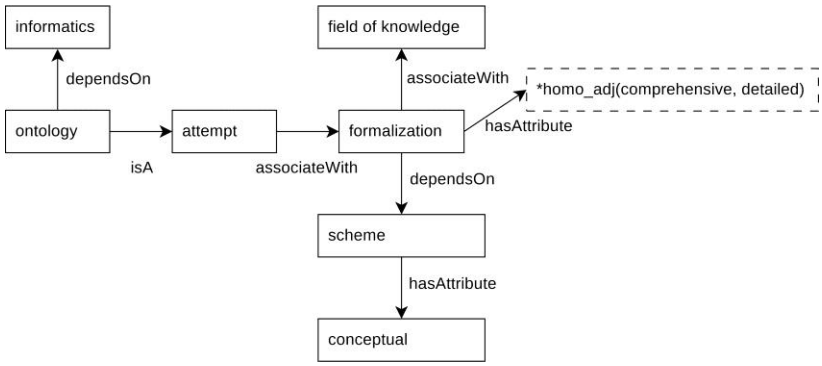

Fig. 5. Example of a semantic tree of test sentence at the third iteration of the algorithm.

As you can see from Figure 5, all syntactic units of the third level of the syntactic tree of the test sentence that not marked as processed were processed. After applying the translation rules:

- form the syntactic unit "***adj_noun**(***homo_adj**(*comprehensive*, *detailed*), formalization)" the node "formalization" and the temporary node "***homo_adj**(...)" were formed that are connected by the relation "hasAttribute". In a pair adjective-noun a noun is the main node, so the existing relationships refers to a noun;
- also between the nodes "attempt" and "formalization" a relation "associateWith" was created.

All syntactic units of the third level and all syntactic units of the fourth level that are related to the syntactic units of the third level were marked as processed in syntactic tree of test sentence.

Figure 6 shows the semantic tree of test sentence at the fourth iteration of the algorithm.

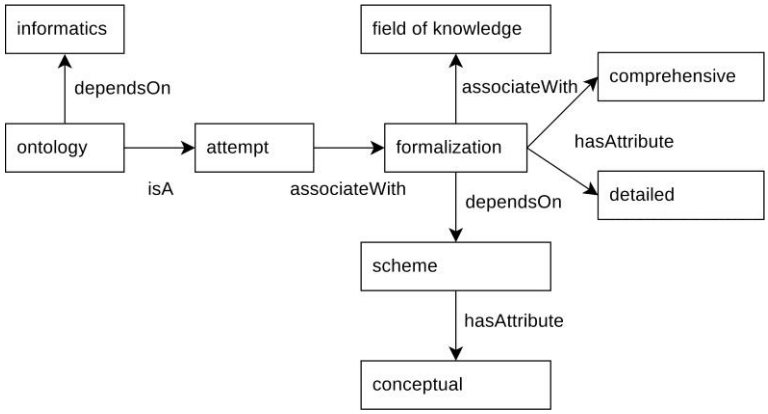

Fig. 6. Example of a semantic tree of test sentence at the fourth iteration of the algorithm.

As you can see from Figure 6, all syntactic units of the fourth level of the syntactic tree of the test sentence that not marked as processed were processed. After applying the translation rules form the syntactic unit "***homo_adj**(*comprehensive*, *detailed*)" the nodes "comprehensive" and "comprehensive" of semantic tree were formed that are connected by relation "hasAttribute" with node "formalization".

All syntactic units of the fourth level and all syntactic units of the fifth level that are related to the syntactic units of the fourth level were marked as processed in syntactic tree of test sentence.

At the fifth iteration of the algorithm, the process of building the semantic tree of the test sentence is complete. The resulting semantic tree for the test fragment is shown in Figure 6. The resulting semantic tree can be merged with other semantic trees in a text resource. In addition, this semantic tree can be merged with the domain ontology created by the expert.

5 Conclusions and Future Work

We have described a modular pipeline that can be used for translating a syntactic tree of sentence into a semantic tree. This approach can be used to automatically build a domain ontology. Manually building an ontology is a long and complex process. The main lack of domain ontologies is the need for their development and updating due to PrA change. The idea of our approach is to use the existing methods of linguistic analysis to construct a syntactic tree of sentence. Further, using a set of rules, you can translate a syntax tree into a semantic tree. Semantic representation of the text on natural language (NL) is the most complete of those that can be achieved only by linguistic methods. The domain ontology can be built from the semantic trees extracted from content of text resources.

Also, we have described the algorithm of translating a syntactic tree into a semantic tree. An example of the proposed approach of translating the syntactic tree of test sentence into a semantic tree is considered in detail.

In the future work we plan to use methods of deep learning to translating the syntactic tree of sentence into a semantic tree. Comparison of the two approaches to solving problem of automatically build a domain ontology will allow us to understand when you need to use the semantic approach and when you need to use the methods of deep learning.

Also, we plan to extend the set of rules for translating the syntactic tree into a semantic tree to cover a greater number of types of semantic relationships between objects of PrA.

In addition, we plan to develop an algorithm for evaluating the quality of the resulting ontology.

Acknowledgments. This study was supported Ministry of Education and Science of Russia in framework of project № 2.4760.2017/8.9 and by the Russian Foundation for Basic Research (Grants No. 16-47-732054 and 18-37-00450).

References

1. Konstantinova, N.S., Mitrofanova, O.A.: Ontology as a knowledge storage system // Portal Information and Communication Technologies in Education. Available at: http://www.ict.edu.ru/ft/005706/68352e2-st08.pdf (accessed: 21.03.2018)
2. Martino, B.D: An Approach to Semantic Information Retrieval Based on Neutral Language Query Understanding. In: 10th International Conference on Web Engineering ICWE 2010 Workshops, pp. 211–222 (2010)
3. Damljanovic, D., Agatonovic, M., Cunningham, H.: Natural Language Interfaces to Ontologies: Combining Syntactic Analysis and Ontology-based Lookup through the User Interaction. Available at: https://gate.ac.uk/sale/eswc10/freya-main.pdf (accessed: 22.03.2018)
4. Pazienza, M., Pennacchiotti, M., Zanzotto, F.: Terminology extraction an analysis of linguistic and statistical approaches. In: NEMIS 2004, pp. 255–279 (2004)
5. Nenadic, G., Ananiadou, S., McNaught, J.: Enhancing Automatic Term Recognition through Variation. In: Conference on Computational Linguistics (COLING-04), pp. 604–610 (2004)
6. Zarubin, A., Koval, A., Filippov, A., Moshkin, V.: Application of syntagmatic patterns to evaluate answers to open-ended questions. In: Communications in Computer and Information Science (CITDS-2017), pp. 150–162 (2017)
7. Zarubin, A.A., Koval, A.R., Moshkin, V.S., Filippov, A.A.: Construction of the problem area ontology based on the syntagmatic analysis of external wiki-resources. Available at: http://ceur-ws.org/Vol-1903/paper26.pdf (accessed: 03.03.2018).
8. Shestakov, V.K.: Development and maintenance of information systems based on ontology and Wiki-technology. In: 13-and All-Russian scientific. conf. "RCDL–2011", Voronezh, pp. 299–306 (2011)
9. Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., Hellmann, S.: DBpedia -- A Crystallization Point for the Web of Data. Journal of Web Semantics: Science, Services and Agents on the World Wide Web, Vol. 7, pp. 154–165 (2009)
10. Suchanek, F.M., Kasneci, G., Weikum, G.: YAGO: A Core of Semantic Knowledge Unifying WordNet and Wikipedia. In: Proceedings of the 16th International Conference on

World Wide Web (Banff, Alberta, Canada, May 8–12, 2007), NY: ACM Press, pp. 697–706 (2007)
11. Suchanek, F.M., Kasneci, G., Weikum, G.: YAGO: A Large Ontology from Wikipedia and WordNet. Web Semantics: Science, Services and Agents on the World Wide Web, Vol. 6, Issue 3, pp. 203–217 (2008)
12. Subkhangulov, R.A.: Ontological search for technical documents based on the intelligent agent model. Automation of management processes 4(38), pp. 85–91 (2014)
13. Astrakhantsev, N.A., Fedorenko, D.G., Turdakov, D.Y.: Automatic Enrichment of Informal Ontology by Analyzing a Domain-Specific Text Collection. In: Materials of International Conference "Dialog" 13(20), pp. 29–42 (2014)
14. Cui, G.Y., Lu, Q., Li, W.J., Chen, Y.R.: Corpus Exploitation from Wikipedia for Ontology Construction. In: Proceedings of the Sixth International Language Resources and Evaluation (LREC 2008), Marrakech, pp. 2125–2132 (2008)
15. Hepp, M., Bachlechner, D., Siorpaes, K.: Harvesting Wiki Consensus – Using Wikipedia Entries as Ontology Elements. In: Proceedings of the First Workshop on Semantic Wikis – From Wiki to Semantics, Annual European Semantic Web Conference (ESWC 2006), pp. 124–138 (2006)
16. McGuinness, D.L., Fikes, R., Rice, J., Wilder, S.: An environment for merging and testing large ontologies. KR, pp. 483–493 (2000)
17. Noy, N.F., Musen, M.A.: Algorithm and Tool for Automated Ontology Merging and Alignment. In: Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence, pp. 450–455 (2000)
18. Pottinger, R., Bernstein, P.A.: Schema merging and mapping creation for relational sources. EDB, pp. 73–84 (2008)
19. Raunich, S., Rahm, E.: ATOM: Automatic Target-driven Ontology Merging. AbteilungDatenbanken der Universität Leipzig. Available at: http://dbs.uni-leipzig.de/file/ATOM-ICDE11_demo_final.pdf (accessed: 22.02.2018).
20. Raunich, S., Rahm, E.: Target-driven Merging of Taxonomies. Cornell University Library. Available at: https://arxiv.org/ftp/arxiv/papers/1012/1012.4855.pdf (accessed: 18.02.2018) (2010)
21. Sokirko, A.V.: Semantic words in automatic processing: dis. Phd (05.13.17); State Committee of the Russian Federation for Higher Education Russian State University for the Humanities, p. 120 (2001)
22. Boyarskiy, K.K., Kanevskiy, Ye.A.: Semantico-syntactic parser SemSin. Scientific and Technical Herald of Information Technologies, Mechanics and Optics, Vol 5, pp.869–876 (2015)
23. Artemov, M.A., Vladimirov, A.N., SeleznevK, Ye.: Survey of natural text analysis systems in Russian. Scientific journal Bulletin of the Voronezh State University. Available at: http://www.vestnik.vsu.ru/pdf/analiz/2013/02/2013-02-31.pdf (accessed: 22.03.2018).
24. AoT: Automatic text processing. Available at: http://aot.ru (accessed: 22.03.2018)