

Article

Building a Domain Ontology in the Process of Linguistic Analysis of Text Resources

Nadezhda Yarushkina, Aleksey Filippov*, Vadim Moshkin and Yuri Egorov

Ulyanovsk State Technical University, 432027, Ulyanovsk, Russia

E-mails: jng@ulstu.ru, al.filippov@ulstu.ru, v.moshkin@ulstu.ru, yupe@mail.ru

* Correspondence: al.filippov@ulstu.ru; Tel.: +7-908-485-8390

Abstract: Ontology is a formalized representation of the problem area (PrA). Representation of the PrA in the form of an 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. This article describes the method of building a domain ontology based on the linguistic analysis of content of text resources, presents an example of the proposed approach, and also presents the architecture of our pipeline.

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 [1–6].

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 lack 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:

- extraction of knowledge from Internet resources (in particular, wiki-resources);
- analysis of dictionaries and thesaurus;
- merging of different ontological structures;
- 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:

- 37 1. Formation of *classes* and relations of ontology on the basis of analysis of the structure of
38 wiki-resources [7–11].
- 39 2. Formation of *objects* and relations of ontology on the basis of analysis of the structure of
40 wiki-resources [7,12–15].
- 41 3. Formation of an ontology in the process of combining several ontologies [16–20].

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

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

55 2. A Method of Translating a Syntactic Tree Into a Semantic Tree

56 It is necessary to determine the syntactic structure of the sentence on NL for constructing the
57 semantic tree. There are several parsing tools of texts in Russian, for example [21–24]:

- 58 • Lingo-Master;
- 59 • Treeton;
- 60 • DictaScope Syntax;
- 61 • ETAP-3;
- 62 • ABBYY Compreno;
- 63 • Tomita-parser;
- 64 • AOT etc.

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

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

72 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:

$$F^{Sem} : \{N_{li}^{Synt}, P_j\} \rightarrow \{N^{Sem}, R^{Sem}\},$$

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

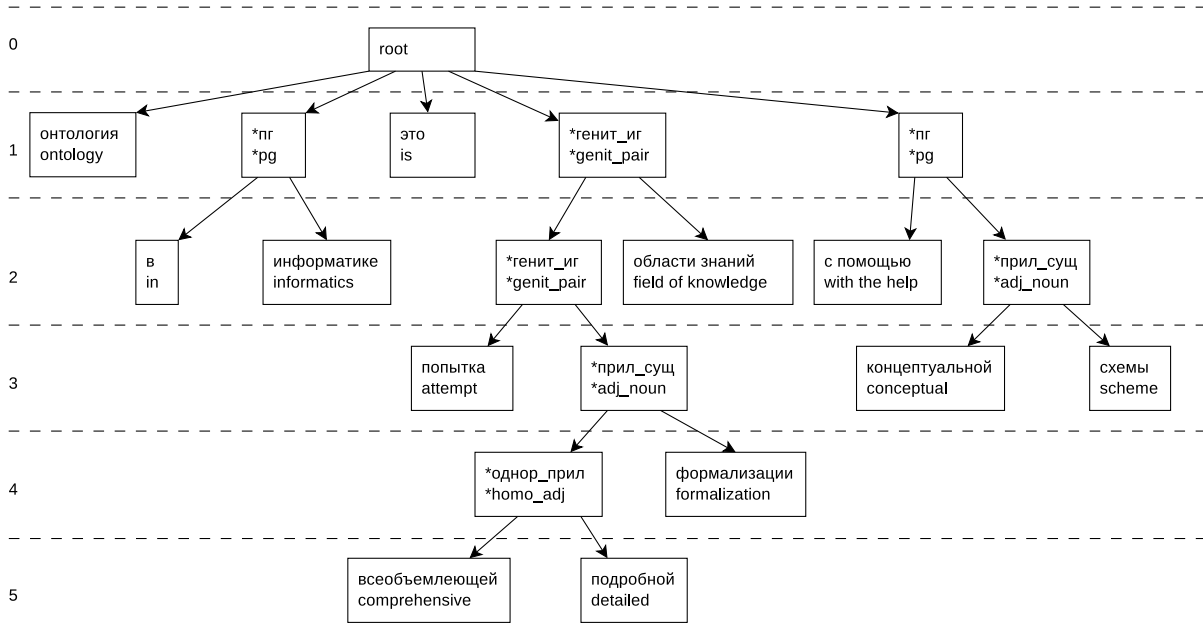


Figure 1. Example of a syntactic tree of test sentence.

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

$$\{N_1^{Synt}, N_2^{Synt}, \dots, N_k^{Synt}\} \rightarrow \{N^{Sem}, R^{Sem}\}, k = \overline{1, K},$$

79 where $\{N_1^{Synt}, N_2^{Synt}, \dots, N_k^{Synt}\}$ is the set of units of the rule corresponding to the set of nodes of
 80 the syntactic tree. The rule only works if all the units match. Examples of rules and the results of
 81 their use are presented in Table 1; K is number of units in the rule; $\{N^{Sem}, R^{Sem}\}$ is set of nodes N^{Sem}
 82 and relations R^{Sem} of the semantic tree, obtained as a result of translation of the syntactic tree into a
 83 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
<i>attempt</i> -*genit_pair- <i>formalization</i>	node1-*genit_pair-node2 → node1-associateWith-node2	<i>attempt</i> -associateWith- <i>formalization</i>
<i>in</i> -*pg- <i>informatics</i>	node1-*pg-node2 → prevNode-dependsOn-node2	lastNode-dependsOn- <i>informatics</i>
is	is → prevNode-isA-nextNode	lastNode-isA-nextNode
<i>conceptual</i> -*adj_noun- <i>scheme</i>	node1-*adj_noun-node2 → node2-hasAttribute-node1	<i>scheme</i> -hasAttribute- <i>conceptual</i>
<i>comprehensive</i> -*homo_adj- <i>formalization</i>	node1-*homo_adj-node2 → node2-hasAttribute-node1	<i>formalization</i> -hasAttribute- <i>comprehensive</i>
<i>detailed</i> -*homo_adj- <i>formalization</i>		<i>formalization</i> -hasAttribute- <i>detailed</i>

Formally R^{Sem} :

$$R^{Sem} = \{R_{isA}^{Sem}, R_{partOf}^{Sem}, R_{associateWith}^{Sem}, R_{dependsOn}^{Sem}, R_{hasAttribute}^{Sem}\},$$

84 where R_{isA}^{Sem} is set of transitive relations of hyponymy; R_{partOf}^{Sem} is set of transitive relations "part/whole";
 85 $R_{associateWith}^{Sem}$ is set of symmetrical relations of association; $R_{dependsOn}^{Sem}$ is asymmetric relations of
 86 associative dependence; $R_{hasAttribute}^{Sem}$ is set of asymmetric relations describing the attributes of nodes.

87 2.1. The Algorithm of Translating a Syntactic Tree into a Semantic Tree

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

- 89 1. Go to the first level of the syntactic tree.
- 90 2. Select the next node of the current tree level. If there are no unprocessed nodes, go to step 12.
- 91 3. If the node is marked as processed, go to step 2.
- 92 4. If the node is not a syntax label (not starts with "*"), go to step 10.
- 93 5. If the node is a syntax label (starts with "*") and does not have child elements, go to step 10.
- 94 6. If the node is a syntax label (starts with "*") and all its child nodes are not syntax labels, go to step
95 10.
- 96 7. If there is a temporary parent node, then replace it, otherwise create a temporary node.
- 97 8. If there is no connection between the nodes, create a temporary relationship between them and
98 go to step 2.
- 99 9. If both nodes are not temporary and there is no connection between them, create an "associateWith"
100 relationship between them and go to step 2.
- 101 10. Apply the rule for translation.
- 102 11. Mark the nodes as processed and go to step 2.
- 103 12. Go to the next level of the syntactic tree, and then go to step 2.

104 2.2. Example of the Algorithm of Translating a Syntactic Tree into a Semantic Tree

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

- 108 • *ontology*;
- 109 • **pg(in, informatics)*;
- 110 • *is*;
- 111 • **genit_pair(*genit_pair(attempt, *adj_noun(*homo_adj(comprehensive, detailed), formalization)),
112 field of knowledge)*;
- 113 • **pg(with the help, *adj_noun(conceptual, scheme))*.

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

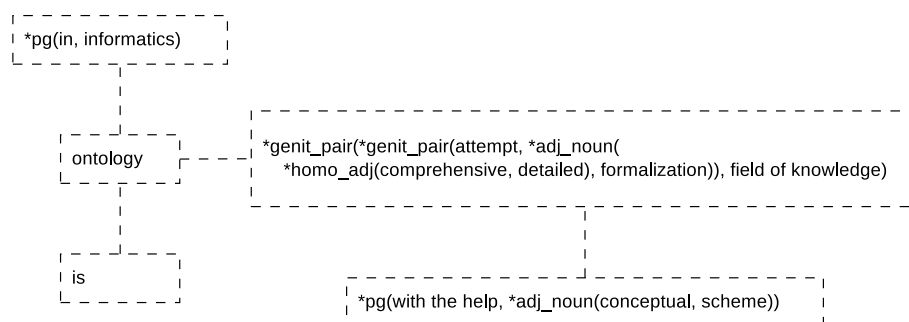


Figure 2. Example of a semantic tree of test sentence at the beginning of the algorithm.

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

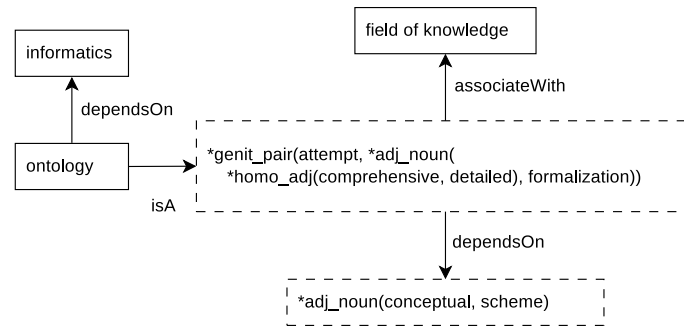


Figure 3. Example of a semantic tree of test sentence at the first iteration of the algorithm.

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

- 118 • the syntactic unit "ontology" was included in semantic tree;
- 119 • from the syntactic unit "is" the relation "isA" was formed between the node "ontology" and the
- 120 temporary node "*genit_pair(...)";
- 121 • from the syntactic unit "*pg(in, informatics)" the node "informatics" and relation "dependsOn"
- 122 between the nodes "informatics" and "ontology" were formed;
- 123 • from the syntactic unit "*genit_pair(*genit_pair(...), field of knowledge)" the temporary node
- 124 "*genit_pair(...)" and the node "field of knowledge" were formed that are connected by the
- 125 relation "associateWith";
- 126 • from the syntactic unit "*pg(with the help, ...)" the temporary node "*adj_noun(conceptual, scheme)"
- 127 and relation "dependsOn" between that node and the temporary node "*genit_pair(...)" were
- 128 formed.

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

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

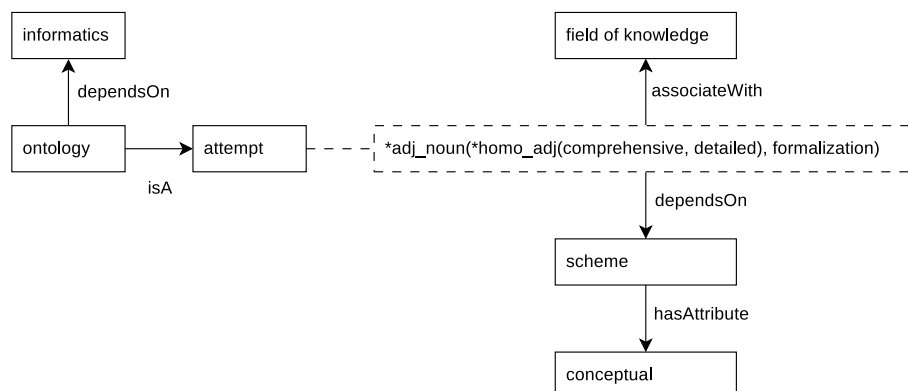


Figure 4. Example of a semantic tree of test sentence at the second iteration of the algorithm.

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

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

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

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

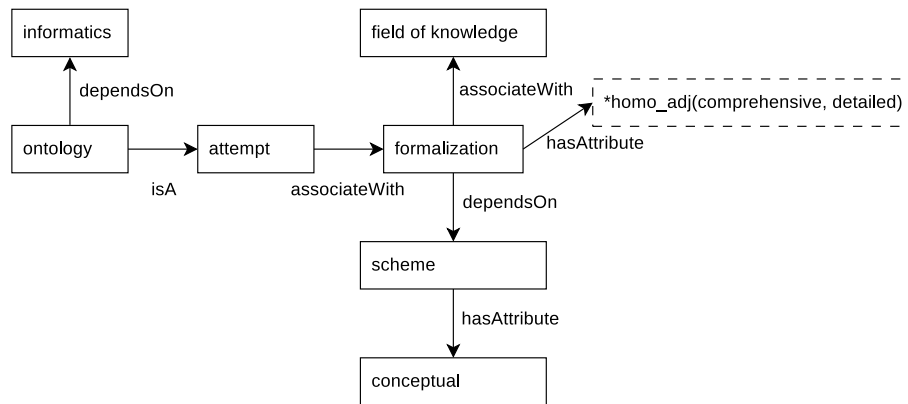


Figure 5. Example of a semantic tree of test sentence at the third iteration of the algorithm.

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

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

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

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

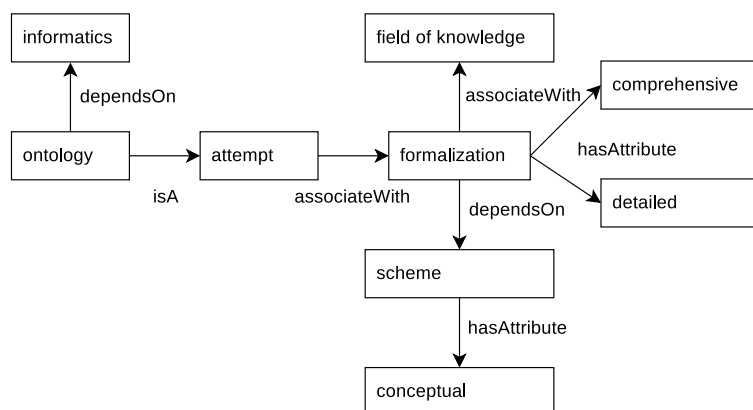


Figure 6. Example of a semantic tree of test sentence at the fourth iteration of the algorithm.

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

157 All syntactic units of the fourth level and all syntactic units of the fifth level that are related to the
 158 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.

2.3. The Architecture of the Pipeline of Translating a Syntactic Tree into a Semantic tree

In our work, the results of the AOT project were used for constructing a syntactic tree. The AOT project started in Moscow in Dialing Company. Russian website is www.aot.ru [24]. The AOT Russian Linguistic Environment (AOT RML) is distributed under the Library GNU Public Licence, that makes it possible to freely use the AOT RML for solving tasks of automated text processing in NL. The AOT RML was written by Alexey Sokirko, Igor Nozhov, Lev Gershenson, Andrey Putrin and many other people.

The source code of AOT RML is available at <https://sourceforge.net/p/seman/svn/HEAD/tree/trunk/>. You can get the installation and configuration manual at <https://sourceforge.net/p/seman/svn/HEAD/tree/trunk/readme>.

Our system is split in two pieces:

- **AOT RML Environment** is the third-party tool that used for grafematic, morphological and syntactic analysis.
- **Semantic Environment** is used for semantic analysis.

Figure 7 is shows the architecture of the pipeline for translating a syntactic tree into a semantic tree.

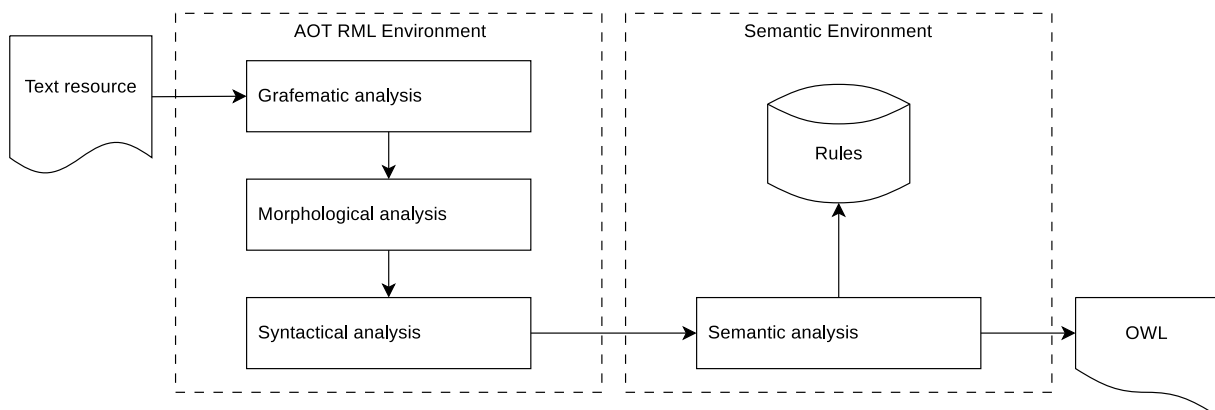


Figure 7. The architecture of the pipeline for translating a syntactic tree into a semantic tree.

The following pipeline of linguistic methods are used to build an ontology based on the analysis of the contents of text resources [21,24]:

1. **Grafematic analysis** is the process of initial analysis of the text on NL. The grafematic analysis presents the input text data in a convenient format for further analysis (separation of input text into words, delimiters etc).
2. **Morphological analysis** is used for construction of morphological interpretation of words of the input text. Morphological dictionary of AOT RML for Russian is based on the grammatical dictionary by A. A. Zaliznyak. The morphological dictionary of AOT RML for the Russian language includes 161 thousand lemmas. In the process of lemmatization, a set of morphological interpretations is generated for each word of the input text: lemma, morphological part of speech, a set of elementary morphological descriptors relating the wordform to some morphological class.
3. **Syntactic analysis** is used for construction of syntactic tree from extracted syntactic groups consisting of simple sentences. Simple sentences can be part of complex ones.

192 4. **Semantic analysis** is used for building the semantic structure of one sentence in Russian. The
193 semantic structure consists of semantic nodes and semantic relations. A set of rules is used for
194 semantic analysis. Each rule uses the results of the syntactic analysis.

195 As a result of the algorithm a set of semantic trees is formed. Thus, resulting semantic trees form
196 the initial (raw/unprocessed) version of the domain ontology. The resulting domain ontology is saved
197 as OWL file.

198 3. Conclusions

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

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

210 In our work, the results of the AOT project were used for constructing a syntactic tree. Our system
211 is split in two pieces: AOT RML Environment and Semantic Environment. AOT RML Environment
212 is the third-party tool that used for grafematic, morphological and syntactic analysis. Semantic
213 Environment is used for semantic analysis. The pipeline of linguistic methods are used to build
214 an ontology based on the analysis of the contents of text resources. The pipeline includes blocks
215 of grafematic, morphological, syntactic and semantic analysis. As a result of the algorithm a set of
216 semantic trees is formed. Thus, resulting semantic trees form the initial (raw/unprocessed) version of
217 the domain ontology. The resulting domain ontology is saved in the OWL file. The generated OWL
218 file can be modified by an expert.

219 3.1. Future Work

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

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

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

227 **Acknowledgments:** This work was financially supported by the Russian Foundation for Basic Research (Grants
228 No. 16-47-732054 and 16-47-730742).

229 **Author Contributions:** All authors have made significant contributions to the paper. Nadezhda Yarushkina
230 and Yuri Egorov carried out a literature survey. Vadim Moshkin and Aleksey Filippov proposed the method of
231 translating a syntactic tree into a semantic tree.

232 **Conflicts of Interest:** The authors declare no conflict of interest.

233 Abbreviations

234 The following abbreviations are used in this manuscript:

235 PrA	Problem area
236 NL	Native language
AOT RML	AOT Russian Linguistic Environment

237 **References**

- 238 1. Konstantinova N.S., Mitrofanova O.A. Ontology as a knowledge storage system //
239 Portal "Information and Communication Technologies in Education". Available at:
240 <http://www.ict.edu.ru/ft/005706/68352e2-st08.pdf> (accessed: 21.11.2017).
- 241 2. Martino B.D. An Approach to Semantic Information Retrieval Based on Neutral Language Query
242 Understanding // 10th International Conference on Web Engineering ICWE 2010 Workshops, 2010, pp.
243 211-222.
- 244 3. Damljanovic D., Agatonovic M., Cunningham H. Natural Language Interfaces to Ontologies:
245 Combining Syntactic Analysis and Ontology-based Lookup through the User Interaction. Available at:
246 <https://gate.ac.uk/sale/eswc10/freya-main.pdf> (accessed: 22.11.2017).
- 247 4. Paziienza M., Pennacchiotti M., Zanzotto F.: Terminology extraction an analysis of linguistic and statistical
248 approaches // NEMIS 2004, 2004, pp. 255–279.
- 249 5. Nenadic G., Ananiadou S., McNaught J.: Enhancing Automatic Term Recognition through Variation //
250 Conference on Computational Linguistics (COLING-04), 2004, pp. 604–610.
- 251 6. Zarubin A., Koval A., Filippov A., Moshkin V. Application of syntagmatic patterns to evaluate answers to
252 open-ended questions // Communications in Computer and Information Science (CITDS-2017), 2017, pp.
253 150–162.
- 254 7. Zarubin A.A., Koval A.R., Moshkin V.S., Filippov A.A. Construction of the problem area ontology based on
255 the syntagmatic analysis of external wiki-resources. Available at: <http://ceur-ws.org/Vol-1903/paper26.pdf>
256 (accessed: 03.11.2017).
- 257 8. Shestakov V.K. Development and maintenance of information systems based on ontology and
258 Wiki-technology // 13-and All-Russian scientific. conf. "RCDL-2011", Voronezh, 2011, pp. 299–306.
- 259 9. Bizer C., Lehmann J., Kobilarov G., Auer S., Becker C., Cyganiak R., Hellmann S. DBpedia – A Crystallization
260 Point for the Web of Data // Journal of Web Semantics: Science, Services and Agents on the World Wide
261 Web, 2009, Vol. 7, pp. 154–165.
- 262 10. Suchanek F.M., Kasneci G., Weikum G. YAGO: A Core of Semantic Knowledge Unifying WordNet and
263 Wikipedia // Proceedings of the 16th International Conference on World Wide Web (Banff, Alberta, Canada,
264 May 8–12, 2007), NY: ACM Press, 2007, pp. 697–706.
- 265 11. Suchanek F.M., Kasneci G., Weikum G. YAGO: A Large Ontology from Wikipedia and WordNet // Web
266 Semantics: Science, Services and Agents on the World Wide Web, Vol. 6, Issue 3, 2008, pp. 203–217.
- 267 12. Subkhangulov R.A. Ontological search for technical documents based on the intelligent agent model //
268 Automation of management processes, 2014, Vol. 4 (38), pp. 85-91.
- 269 13. Astrakhantsev N.A., Fedorenko D.G., Turdakov D.Y. Automatic Enrichment of Informal Ontology by
270 Analyzing a Domain-Specific Text Collection // Materials of International Conference "Dialog", 2014, Vol.
271 13 (20), pp. 29–42.
- 272 14. Cui G.Y., Lu Q., Li W.J., Chen Y.R. Corpus Exploitation from Wikipedia for Ontology Construction //
273 Proceedings of the Sixth International Language Resources and Evaluation (LREC 2008), Marrakech, 2008,
274 pp. 2125–2132.
- 275 15. Hepp M., Bachlechner D., Siorpaes K. Harvesting Wiki Consensus – Using Wikipedia Entries as Ontology
276 Elements // Proceedings of the First Workshop on Semantic Wikis – From Wiki to Semantics, Annual
277 European Semantic Web Conference (ESWC 2006), 2006, pp. 124–138.
- 278 16. McGuinness D.L., Fikes R., Rice J., Wilder S. An environment for merging and testing large ontologies //
279 KR, 2000, pp. 483–493.
- 280 17. Noy N.F., Musen M.A. Algorithm and Tool for Automated Ontology Merging and Alignment // Proceedings
281 of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative
282 Applications of Artificial Intelligence, 2000, pp. 450-455.
- 283 18. Pottinger R., Bernstein P.A. Schema merging and mapping creation for relational sources // EDB , 2008, pp.
284 73–84.
- 285 19. Raunich S., Rahm E. ATOM: Automatic Target-driven Ontology Merging // Abteilung Datenbanken der
286 Universität Leipzig. Available at: http://dbs.uni-leipzig.de/file/ATOM-ICDE11_demo_final.pdf (accessed:
287 22.02.2017).

- 288 20. Raunich S., Rahm E. Target-driven Merging of Taxonomies // Cornell University Library, 2010. Available at:
289 <https://arxiv.org/ftp/arxiv/papers/1012/1012.4855.pdf> (accessed: 18.02.2017).
- 290 21. Sokirko A.V. Semantic words in automatic processing: dis. Phd (05.13.17); State Committee of the Russian
291 Federation for Higher Education Russian State University for the Humanities, 2001, p. 120.
- 292 22. Boyarskiy K.K., Kanevskiy Ye.A. Semantico-syntactic parser SemSin // Scientific and Technical Herald of
293 Information Technologies, Mechanics and Optics. vol.5. 2015. pp.869-876.
- 294 23. Artemov M.A., Vladimirov A.N., Seleznev K.Ye. Survey of natural text analysis systems
295 in Russian // Scientific journal Bulletin of the Voronezh State University. Available at:
296 <http://www.vestnik.vsu.ru/pdf/analiz/2013/02/2013-02-31.pdf> (accessed: 22.11.2017).
- 297 24. Automatic text processing // Automatic text processing. Available at: <http://aot.ru> (accessed: 22.11.2017).