Approach to automatic population of ontologies of scientific subject domain using lexico-syntactic patterns

Yu A Zagorulko, E A Sidorova, I R Akhmadeeva and A S Sery
A.P. Ershov Institute of Informatics Systems, Siberian Branch of the Russian Academy of Sciences, 6, pr. Acad. Lavrentieva, 630090, Novosibirsk, Russia
E-mail: zagor@iis.nsk.su

Abstract. This paper presents an approach to automatic population of ontologies of a scientific subject domain (SSD) using Lexico-Syntactic Patterns (LSPs) and a corpus of texts related to modeled domain. The main feature of this approach is that such patterns are automatically built based on Ontology Design Patterns of other types provided by the system for the automated development of SSD ontologies using heterogeneous Ontology Design Patterns. The implementation of the ontology population using constructed LSPs is described in detail. The results of the experiments on the SSD ontology population are presented. It is noted that there is a problem in establishing a subject of a relation when extracting facts. To address this problem, the authors are planning to employ the coreference resolution methods.

1. Introduction
Currently, ontologies are recognized as the most effective means of formalizing and systematizing knowledge and data in the scientific subject domains (SSDs). The process of developing a domain ontology consists of several stages, the main of which are the construction of the terminological part of the ontology, which implies the creation of taxonomies of concepts and relations and the description of their properties, and the population of the ontology, i.e. adding instances of concepts and relations to it. If the first stage defines the skeleton of the ontology, then the second stage fills it with content.

To obtain an ontology that would describe SSD in considerable detail, it is required to process a huge number of publications and information resources containing information from the modeled area. To facilitate and accelerate this process, methods of automatic ontology population based on natural language texts [1, 2] and web documents [3, 4] are being developed [5, 6]. Lexico-Syntactic Patterns (LSPs) are also used to populate ontologies [7]. Such LSPs map the language structures found in the text onto the corresponding elements of the ontology (concepts, relations, instances of concepts and relations).

This paper describes an approach to the automatic population of the SSD ontologies using Lexico-Syntactic Patterns. Unlike the LSPs used in [7], the LSPs used in our approach are automatically built based on Ontology Design Patterns (ODPs) of other types [8] included in the system for the automated construction of the SSD ontologies using heterogeneous Ontology Design Patterns [9].
The rest of the paper is organized as follows. The second Section presents the language for describing the LSPs. The third Section describes the principles of the automatic generation of the LSPs based on patterns of other types of the ODPs, a general scientific dictionary and the current version of ontology. The fourth Section deals with the implementation of the ontology population using the constructed LSPs. In this Section, the results of the experiments on the ontology population based on the generated LSPs are presented. In Conclusion, the results of the implementation and the use of this approach are summarized, and the prospects for its development are outlined.

2. Representation of Lexico-Syntactic Patterns

The LSP language allows us to define a multilevel system of Lexico-Syntactic Patterns, consisting of Terminological Patterns (T-LSP) describing elementary language constructions and lexico-syntactic “samples” for their extraction, and Information Lexico-Syntactic Patterns (I-LSP) defining a scheme for extracting facts from a text and generating the corresponding elements of ontology.

We consider facts as triples of the form $\langle O, P, V \rangle$, where $O$ is an object (an individual or an instance of some ontology class), $P$ is either a “type” relation linking individual to class, or an ontology relation label (ObjectProperty), or an ontology attribute label (DatatypeProperty), $V$ is either an ontology class, an instance of a class, or a simple value of a standard type.

Each LSP implements a model of the form $\langle Args, Constrs, Res \rangle$, where $Args$ are either domain terms or objects (provided that objects have been already extracted using other I-LSPs). $Constrs$ are semantic and/or syntactic constraints on $Args$, and $Res$ describes the result of LSP matching, which can be either a new term (for T-LSP) or an ontology fragment (for I-LSP).

Note that the automatic ontology population using LSPs requires extracting the terms that are specific for a given SSD from the text. To this end, we can employ both a domain-specific dictionary and specialized Terminological Patterns. The patterns allow for extracting the terms not included in the dictionary (in particular, the names of SSD objects).

A domain-specific dictionary is a lexicon organized according to the semantic principle, reflecting a certain set of basic formal relations. In our approach, the domain-specific dictionary stores the necessary information both for extracting a term from the text and for supporting the subsequent stages of the analysis.

Formally the dictionary is described as a system of the form $D = \langle W, P, M, G, S \rangle$, where $W$ is a set of lexemes, each lexeme associated with the information about the entire set of its forms; $P$ is a set of multi-word terms, each of which is a pair $\langle N$-gram, structure type $\rangle$, where $N$-gram sets a sequence of lexemes and structure type defines the head and rules of agreement of the $N$-gram elements; $M$ is a morphological model of a language including the description of morphological classes and features; $G$ is a set of rules of agreement for extracting multi-word terms; $S$ is a problem-oriented system of lexico-semantic features.

Each dictionary term found in the text is supplied with morphological and semantic information, which is further used in the LSP matching.

2.1. Terminological Patterns

Terminological Patterns describe elementary language constructions and lexico-syntactic ”samples” for their extraction.

Terminological Patterns (T-LSPs) are used to retrieve the domain terms that are not predefined in the dictionary. They are formed on the basis of reference terms, markers, semantic and syntactic constraints. Three groups of such patterns are proposed.

The patterns of the first group are utilized for extracting labels of individuals based on a head word or a term using syntactic rules for the noun phrases identification.
\[ \langle Adj \rangle^*, X, \langle Adj \rangle^*, \langle N, GEN \rangle^* \] \rightarrow X.Name \tag{1} 

This generic pattern allows us to generate specific T-LSPs by substituting the ontology class labels instead of \( X \). An example of substitutions is T-LSP (7).

The patterns of the second group are formed using relation indicators, which, as a rule, are predicates from the dictionary, matched with attribute labels and relation labels of Content ODP, or predicates extracted from the Competency questions (see Section 3).

\[ \langle X.Name \rangle, Y\langle Rel \rangle, \mathbf{St}\langle NP^* \rangle \] \rightarrow Z.Name \tag{2} 

Based on these generic patterns, we can generate T-LSPs in which the class \( X \) has a property (Object Property in the first case, Datatype Property — in the second) and assign the semantics of the class \( Z \) or attribute of the class \( X.A \) to the extracted name groups.

The patterns of the third group are used to extract new predicates intended for the search for relations (we consider those predicate terms that serve to extract the same relation as conditional synonyms).

\[ \langle X.Name \rangle, r\langle VP \rangle, \langle Y.Name \rangle \] \rightarrow X.Rel \tag{3} 

This generic pattern allows us to generate T-LSPs, in which the classes \( X \) and \( Y \) are connected in the ontology by \( \text{Rel} \) relation and to assign the semantics of the corresponding ontological relation to the extracted terms (verb groups). For the example see (8).

2.2. Information Patterns

The I-LSPs are used to extract facts and to create new objects to populate the ontology. Three types of the I-LSPs are proposed.

The I-LSPs of the first type (also called the initializing I-LSPs) create objects based on the terms that have a lexico-semantic feature matching either a class label or a label of the key attribute of that class.

\[ \langle X \rangle \Rightarrow \text{create } X() \] 
\[ \langle X.Name \rangle \Rightarrow \text{create } X(\text{Name} : \text{arg1}) \tag{4} \]

The generic patterns (4) allow us to generate (by substituting the ontology class labels instead of \( X \)) specific I-LSPs with which objects will be subsequently extracted (see (9) and (10)).

The I-LSPs of the second type are designed to extract relations between objects (Object Property) based on predicate words. This type of a pattern is described by three arguments: two objects of corresponding classes and a term connecting them.

\[ [X, (X.Rel), Y] \Rightarrow \text{set } X(\text{Rel} : Y) \tag{5} \]

This generic pattern allows us to generate I-LSPs, in which the classes \( X \) and \( Y \) are connected in the ontology with \( \text{Rel} \). The pattern sets the general structure of I-LSP, which usually needs to be specified by a set of syntactic constraints, depending on the predicate term. These constraints can be derived from a corpus of texts, using relations already established at the ontology level, or on the basis of statistical criteria (see (11)).

The next type of I-LSP populates object attributes (Datatype Property) either on the basis of predicate terms signaling the presence of its attribute next to the object, or on the contact way of expressing the attribute value.
\[ [X, \langle Prop.A \rangle, (X.A)] \Rightarrow set X(A : arg2) \]
\[ [X, \langle X.A \rangle] \Rightarrow set X(A : arg2) \] (6)

These generic patterns allow generating I-LSPs that populate an attribute of the object \( X \) with the value of a term with the semantics \( X.A \). Predicate terms with semantics \( Prop.A \) are either attribute labels or verb groups extracted from Competency questions. These patterns can also, as in the previous case, be refined by syntactic constraints.

3. The LSP generation

As was mentioned above, Lexico-Syntactic Patterns are automatically built on the basis of other types of Ontology Design Patterns included in the system for automated construction of the SSD ontologies using heterogeneous Ontology Design Patterns. This system also includes ODPs of the following types: Content Patterns, Structural Logical Patterns, and Presentation Patterns. The latter are subdivided into Naming Patterns and Annotation Patterns.

The Ontology Design Patterns are systematized in the ontology of ODPs according to the types of ontological modeling problems they solve (see figure 1).

The description of the properties of the ODPs is based on the format proposed on the ODPA association portal [10]. In accordance with it, the description of a pattern includes information about its author and scope, its textual description, graphical representation, links to other patterns, a set of scenarios and examples of the use. Content Patterns are additionally supplied with a set of Competency questions reflecting its content [11].

Between the Content Patterns, the relations "generalizes" and "specializes" defining a hierarchy of "general-particular" on them, are set. The "uses" relationship links Structural Logical Patterns to the Content Patterns in which they are used. The relation "corresponds" is also introduced, it links Lexico-Syntactic Patterns with Content Patterns and Structural Logical Patterns on the basis of which they were built.

The Content Patterns, specifying the ways of representing typical fragments of ontologies describing complex entities of SSD, play an important role in the considered system for constructing the SSD ontologies. In particular, the system includes Content Patterns that serve to define such concepts as Research Object, Research Subject, Research Method, Science Section, Scientific Result, Scientific Activity, Project, Person, Organization, Publication, Information Resource, etc.

To represent the Content Patterns and complex entities and structures relevant in the construction of the SSD ontologies, the system includes Structural Logical Patterns of the ODPs, which serve to represent areas of admissible values, n-ary and attributed relations (binary relations with attributes) [12].

To increase the readability of the ontology and ease of its use, as well as to customize it for various types of the users, the system includes Presentation Patterns. The names of ontological elements (in one or several natural languages) are specified in the Presentation Patterns using labels, denoting the names of entities in various texts and terminological systems. These names can further be used to recognize the corresponding entities by Lexico-Syntactic Patterns.

To implement the automatic population of the ontology, it is necessary for each Content Pattern to build a set of Lexico-Syntactic Patterns describing various ways of presenting information corresponding to this pattern in scientific texts. To this end, the description of the Content Pattern in the OWL language [13], the Competency questions developed for it, and the corresponding fragment of the SSD ontology are used.

The description of each Content Pattern presents the names of all attributes and relations that connect the class it describes with standard data types and other classes of the ontology, as well as with a list of the key attributes of the class. Attribute and relation names are represented by language expressions, which are markers introducing relations in the text. The Competency
Figure 1. A simplified ontology of ODPs.

questions, expressed in a natural language, allow one to set initial syntactic restrictions on the extracted facts, which can be subsequently refined on the basis of the text corpus. The information about the key attributes of the class is necessary to generate new objects based on the values of these attributes extracted from the text.

The process of generating Lexico-Syntactic Patterns includes building a subject dictionary, generating Terminological and Information Patterns. When building a subject dictionary, the following actions are performed:

- generation of a set of lexical and semantic features of the dictionary based on hierarchical and structural links between classes of the SSD ontology;
- extraction of all names from the patterns and ontology, their normalization and addition to the subject dictionary with the corresponding lexical and semantic features.

At the stage of building the T-LSP, the following tasks are solved:

- generation of T-LSP for the assembling noun phrases, whose tops are the names of the classes (or their synonyms / quasi-synonyms);
- extraction of predicate terms from Competency questions and their coordination with the terms of the general scientific dictionary (this expands the list of synonyms and quasi-synonyms of predicates presented in the dictionary);
- generation of T-LSP based on the structure of the Content Pattern.

The I-LSP generation includes:

- preparation of the I-LSP template for each type of connection in the structure of the Content Pattern;
- filling in the I-LSP argument structure with the corresponding types of related entities and terms;
– clarification of syntactic and positional constraints of the I-LSP on the basis of examples of the I-LSP occurrences in the text corpus.

4. Populating the ontology using the LSPs
The ontology population is the process of inserting the concept and relation instances into an existing ontology [5].

An LSP-based ontology populating system implements a text processing pipeline having the following main stages:

(i) Vocabulary analysis. At this stage, a text undergoes a morphological analysis, and vocabulary terms are extracted from it.

(ii) Extracting new terms. At this stage, we use the T-LSPs to search for the text for contexts, where new terms we do not have in our vocabulary yet might occur. These terms, if found, are added to the vocabulary.

(iii) The last of the main stages is extracting information from the text. The I-LSPs are utilized to combine the terms and other information we have obtained during the previous stages into the objects of the subject domain.

When developing the system, we used tools and technologies we have already developed: the KLAN system, which is a vocabulary extraction tool, the Pattern-based Term Extractor (PatTerm) [14] and FATON, which is the fact assembling tool.

4.1. The vocabulary analysis
The KLAN system facilitates a vocabulary analysis. It is aimed at morphological and surface-syntactic text processing, building domain-specific dictionaries and terms extraction. We used the KLAN system to create a domain-specific dictionary for Russian language. The dictionary is populated with terms obtained during the automatic text processing and the LSP generating procedure. The dictionary contains semantic data in the form of lexico-semantic features. The hierarchy of the features is analogous to the entities of the domain ontology.

The lexico-semantic features form the basis of the ontology-based typification of terms in the dictionary. To the Classifier terms and to the terms that match the names of the classes or datatype properties of the domain ontology were assigned the lexico-semantic features corresponding to the classes or attributes (the names of the latter are prepended with Prop keyword). For example: Method, Author, Task, Object, Result are class-related, while Prop.Date and Prop.Description relate to the datatype properties.

To the terms representing the values of the datatype properties were assigned the features of a form \((\text{ClassName}.\text{PropertyName})\), for example, Method.Name, Method.Description, Method.Date etc. Similarly, the terms that describe relations have the features of the form \((\text{ClassName}.\text{RelationName})\): Method.is_used_for, Method.is_implemented_in etc.

The lexico-semantic features are assigned to the terms automatically based on the corresponding Content ODPs and Competency Questions.

The hierarchy of the lexico-semantic features was also enriched with the features that are common for the scientific papers and texts. The general features were picked by experts and are independent of the subject domain. For example: Comprehension, Existence and Mental-Speech are general features (figure 2).

As is shown in figure 2, the term ‘use’ is semantically ambiguous and could either be a part of the ‘is_used_in’ or ‘is_implemented_in’ relations. The former is a link between Method and Science Section classes. For example: Use of the Monte Carlo methods started at the end of the 1950s. The second relation ‘is_implemented_in’ links Method with the Scientific Result: It was found that the mathematical method could be used to derive qualitative dependencies... The same term can be used in LSP based on the universal lexico-semantic class Use.
4.2. Extracting the terms that are not in a dictionary
The PatTerm system facilitates T-LSP-based term extraction along with constructing the new T-LSPs given a set of literals.

To validate the approach proposed we collected a small corpus consisting of nine scientific papers and two bachelor theses in Russian language. The total volume of the corpus is 53.9 thousand tokens. We have picked the Content ODP Method and a part of the ontology of Scientific Domain “Decision Support in weakly formalized areas” to create a domain-specific dictionary and the LSPs.

When searching for the names of the individuals of the class Method we used the T-LSP built by instantiating the generic pattern (1) with the name of the class Method (7). The resulting T-LSP allows for extracting maximal-length noun phrases, whose heads are words or phrases having the lexico-semantic feature that match the name of the class (i.e. "Method").

\[
[(\text{Adj})^*, (\text{Method}), [(\text{Adj})^*, (\text{N}, \text{GEN})]^*] \Rightarrow \text{Method.Name}
\]  

(7)

There were 111 occurrences of the T-LSP (7) found in the texts of the corpus and 73 terms, to which the semantic feature Method.Name was assigned. Out of these 73 terms 70 turned out not to be in the dictionary, i.e. they were new terms. Examples are Questionnaire method,
Median Ranks method, Neural Network method, Analytic Hierarchy Process (in Russian it has a word 'Method' in its name), Interview method, Self-assessment method etc.

In a similar manner the T-LSPs were used when extracting the names of individuals of the other classes. The T-LSP created for the class Task was found 42 times, 33 terms were found, and 30 of them were the new terms. For example Cognitive modeling, Ordinal Classification problem, Integer Programming problem etc. However, there were noun phrases incorrectly extracted as the names of individuals. In Russian, such phrases, for example Master thesis task, are syntactically similar to the correct names. The precision of extracting names of individuals using these T-LSPs is 75.3%.

The T-LSPs (8) built on the basis of pattern (3) were used to extract new predicate terms.

\[
[(\text{Method.Name}), x(VP), (\text{Task.Name})] \Rightarrow \text{Method.Solves}
\] (8)

This T-LSP occurred in the texts of the corpus multiple times and produced the following terms: allows for constructing, allows for selecting, finds solution, allows for solving.

4.3. Extracting facts and populating an ontology

The FATON system supports the last stage of the text processing extracting information about objects of the subject domain and populating the domain ontology. Given a set of rules and a set of terms that were extracted from a text, the FATON system assembles facts, which are then used to construct objects of the subject domain. According to the approach proposed, rules are the I-LSPs, and terms are extracted by the KLAN and PatTerm systems.

The FATON implements a multiagent algorithm. There are two types of agents defined depending on what kind of action they are aimed to perform. Information agents relate to the terms and objects, where objects are constructed by the algorithm as it proceeds. The rule agents correspond to the I-LSPs. Thus, the information agents provide the input data to the rule agents, while the rule agents form the tuples of arguments from the input data provided by the information agents, and apply the rules to these tuples, assembling new facts and, therefore, building the new fragments of the domain ontology.

The algorithm starts with initializing the rule agents and information agents, respectively, to the I-LSPs and terms having passed as an input. Each information agent, either a term or an object, right after it is initialized immediately sends its data to all rules it could be an argument of. The data received from the information objects form a queue of arguments within a rule agent. The rule agents process their queues, composing tuples of the size \( n \), where \( n \) is the arity of a rule. When a rule agent applies the rule to a tuple of arguments it removes them from the queue, regardless of whether the rule was applied successfully or not. The rule is applied successfully if all constraints defined by the rule hold. In this case a rule agent updates an existing information agent or creates a new one. If an information agent was updated it resends data to all matching rules.

The algorithm stops if all rule agents have no arguments in their queues, and none of the information agents have new data that could be sent to the rule agents. The algorithm returns a set of objects of the ontology of the subject domain.

To incorporate the constructed objects into the ontology we must compare them with the individuals that the ontology already contains. We call this procedure the identification of objects. Depending on the result of the comparison each object could be inserted into the ontology as a new individual of the corresponding class, be used to update an existing individual or be discarded as an incomplete one [15].

Let us consider an example of the I-LSP-based extraction of objects. Figure 3 shows an example of extracting a relation between the objects of classes Method and Task from the following sentence: The ORKLASS method allows building a complete consistent ordinal classification of all sets of estimates by basic indicators. To address the task, the following set of
The ORKLASS method allows building a complete consistent ordinal classification of all sets of estimates by basic indicators.

**Figure 3.** The LSP-based text processing (in Russian).

I-LSPs was created based on patterns (4) and (5). Both terms ORKLASS Method and Ordinal classification are in the domain-specific dictionary having lexico-semantic features Method.Name and Task.Name. According to I-LSPs (9) and (10), individuals of the corresponding classes were created. The relation Method.Solves between these individuals was established by I-LSP (11).

![Diagram](image)

\[
\begin{align*}
\langle \text{Method.Name} \rangle & \Rightarrow \text{create Method(\text{Name} : \text{arg1})} & (9) \\
\langle \text{Task.Name} \rangle & \Rightarrow \text{create Task(\text{Name} : \text{arg1})} & (10) \\
\langle \text{Method,Solves}, \text{Task} \rangle & \Rightarrow \text{set arg1(\text{Solves} : \text{arg3})} & (11)
\end{align*}
\]

The relation Method.Solves was found 13 times in the texts of the test corpus by matching I-LSP (11). Additionally, in 16 cases I-LSP (11) being matched only partially, thereby the algorithm failed to establish the subject of the relationship while the predicate and the object were established correctly. We are planning to utilize the coreference resolution methods to improve the algorithm [16].

5. Conclusion

This paper proposes an approach to automate the population of ontologies of the scientific subject domains using lexico-syntactic ODPs. A feature of the approach proposed is that the LSPs are automatically constructed based on the other ODPs included in the system for automated construction of ontologies.

We present the LSP language, describe the principles and the algorithm of automatic generation of Lexico-Syntactic Patterns, using ODPs of other types, the general scientific dictionary and the current version of the ontology. The implementation of the ontology population using the constructed LSPs is described in detail. The results of the experiments on the SSD ontology population are presented. It is noted that there is a problem in establishing a subject of a relation when extracting facts. To address this, problem the authors are planning to employ the coreference resolution methods.

Acknowledgments

The research has been supported by Russian Foundation for Basic Research (project no. 19-07-00762).
References

[1] Ijntema W, Sangers J, Hogenboom F and Frasincar F 2012 A lexico-semantic pattern language for learning ontology instances from text J. of Web Semantics 15 27–50
[2] Ruiz-Martínez J, Valencia-García R, Martínez-Béjar R and Hoffmann A 2012 BioOntoVerb: A top level ontology based framework to populate biomedical ontologies from texts Knowledge-Based Systems 36 68–80
[3] Alani H, Kim S, Millard D, Weal M, Hall W, Lewis P and Shadbolt N 2003 Automatic Ontology-Based Knowledge Extraction from Web Documents IEEE Intel. Syst. 18(1) 14–21
[4] Akhmadeeva I, Zagorulko Y and Mouromtsev D 2016 Ontology-Based Information Extraction for Populating the Intelligent Scientific Internet Resources Proc. Int. Conf. on Knowledge Engineering and the Semantic Web (Prague) vol 649, ed A C Ngomo and P Kremen (Cham: Springer International Publishing) 119–128
[5] Petasis G, Karkaletsis V, Palouras G, Krithara A and Zavitsanos E 2011 Ontology population and enrichment: state of the art LNCS 6050 134–166
[6] Ganino G, Lembo D, Mecella M and Scafoglieri F 2018 Ontology population for open-source intelligence: a GATE-based solution Software: Practice and Experience 48(12) 2302–30
[7] Maynard D, Funk A and Peters W 2009 Using lexico-syntactic ontology design patterns for ontology creation and population CEUR Workshop Proc. 516 39–52
[8] Blomqvist E, Hammar K and Presutti V 2016 Engineering ontologies with patterns: the eXtreme design methodology Ont. Engineering with Ont. Design Patterns 25 23–50
[9] Zagorulko Y, Borovikova O and Zagorulko G 2020 The development of a system for automated ontology building based on heterogeneous ontology design patterns J. Phys.: Conf. Series 1715 012014
[10] Association for Ontology Design & Patterns, http://ontologydesignpatterns.org, last accessed 2021/04/30.
[11] Karima N, Hammar K and Hitzler P 2017 How to document ontology design patterns Adv. in Ont. Design and Patterns 32 15–27
[12] Zagorulko Y, Borovikova O and Zagorulko G 2018 Development of ontologies of scientific subject domains using ontology design patterns CCIS 822 141–156.
[13] Antoniou G, Harmelen F 2009 Web Ontology Language: OWL Handbook on Ontologies ed Staab S and Studer R (Berlin: Springer Verlag) 91–110
[14] Sidorova E and Akhmadeeva I 2019 The software environment for multi-aspect study of lexical characteristics of text CEUR Workshop Proc. 2523 306–15
[15] Zagorulko Y, Akhmadeeva I and Sery A 2015 An automatization of collection of information about scientific activity for subject intelligent scientific internet resources CEUR Workshop Proc. 1536 62–68
[16] Garanina N, Sidorova E and Sery A 2018 Multiagent approach to coreference resolution based on the multifactor similarity in ontology population Programming and Computer Software 44(1) 23–34