Using Ontology-Driven Methods to Develop Frameworks for Tackling NLP Problems

Taisiya Kostareva, Svetlana Chuprina, Alexander Nam

Perm State University, 15 Bukireva st., Perm, 614068, Russian Federation
tais@nevod.ru, chuprinas@inbox.ru, alxnam@gmail.com

Abstract. In this paper, we present the meta-tooling framework named TAISim that can be used both as a developer's tool for creating NLP systems and as a NLP learning environment, which allows helping students to construct NLP systems by example in a flexible way. TAISim enables the end user to combine different components of a typical NLP system in order to tackle specific NLP problems. We use ontology-engineering methods to accumulate meta-knowledge about the system construction and about users' activities to control the process of development and using the NLP system. Thanks to ontology-driven methods TAISim can be modified and enriched with additional information resources and program modules by means of a high-level interface. Additionally, we demonstrate how the using of meta-ontology helps us to improve TAISim to tackle ontology design automation problems.

Keywords: Natural Language Processing, Learning environment, NLP system framework implementation, Ontology-driven methods, Ontology extraction methods

1 Introduction

Nowadays one of the pressing problems is effective and high-quality processing of unstructured and semi-structured data presented as a natural language text. To gain expertise and improve skills in developing NLP (Natural Language Processing) systems it is crucially important to design a new type of high-level framework tools, which automate both NLP learning and NLP systems designing. Their environment should be adaptable to personal preferences and needs. The problem is complicated by the fact that NLP problems are various, for example, there are different problems in text mining, speech synthesis and recognition, semantic context search, machine translation areas. In spite of this, usually different NLP systems include common steps of text processing.

To tackle NLP learning problems it is important to provide a step by step demonstration of the related results of each module of the text processing system and to have an opportunity to replace some program components (processing resources) and/or to change some supporting information resources (language resources, data resources and so on). This allows comparing the NLP results obtained for the same input texts

but with the using of different supporting resources. It is also important to adapt NLP system development to tackling the specific text processing problems.

There are a number of freeware NLP tools, which is intended for the purposes mentioned above, for example, OpenNLP[1], Natural Language Toolkit (NLTK)[2], GATE[3], Stanford NLP[4], etc. However, unlike them, the main goal of our framework is the research and usage of higher-level ontology based graphical tools for enrichment/replacement of its components and information resources. Thanks to high-level interface, the created platform will allow the qualified users to expand the range of lexical and syntactic patterns, and even novice users will be able to conduct experiments, to reconstruct NLP system and to expand the existing vocabularies by new concepts. We plan to deliver a broad series of experiments to expand the set of patterns to solve problems for texts in Russian.

In one way or another, any NLP system has the components for the following steps of analysis:

- Tokenization, which is a preprocessing phase intended for tokens creation from an input text. It closely cooperates with the lemmatizer. Each token carries NL graphemes or individual signs consisting of other signs (numbers, non-native language graphemes, punctuation). Graphemes are the smallest semantically distinguishing units (the basic linguistic units) in a written language.
- Morphological analysis, which is intended for the internal structure of words analysis and deals with morphemes (the minimal units of linguistic form and meaning), and how they make up words. In a written language, morphemes are composed of graphemes, or the smallest units of typography. A lexical morpheme is one that has meaning by itself, while a grammatical morpheme specifies the relationship between other morphemes.
- Syntactic analysis, which is intended for identification of syntactic relationships between words in a sentence, the construction of the syntactic structure of sentences.
- Semantic analysis, which is intended for identification of semantic relationships between words and syntactic groups, extracting semantic relations (it is the study of the meaning of linguistic utterances).

It should be stressed that there are different approaches to implementing every kind of analysis listed above and different information resources are used in every phase of a NLP. We have developed a meta-tooling framework named TAISim that includes freeware Serelex (http://serelex.cental.be/) components to perform all kinds of analysis mentioned above with a demonstration of all the intermediate results and the supporting resources, and special visual components are developed by the authors of this paper to help explore different methods and tools for semi-automatic ontology construction and refinement. We use the term “semi-automatic ontology engineering” as

[1] http://opennlp.apache.org/index.html
[2] http://www.nltk.org/
[3] https://gate.ac.uk/
[4] http://nlp.stanford.edu/

opposed to ontology learning to emphasize the methodological and interactive aspects of ontology extracting even from a single NL text (not only from corpora) to help domain experts and ontology engineers as well as students to build better and more reasonable ontologies.

Similarly to customizable expert system shells, any TAISim component may be replaced with another component that performs the same functionality and has the same inputs/outputs due to a high-level special control mechanism based on the ontology engineering methods.

2 TAISim as a Learning Environment Tool

As mentioned above, TAISim can be used both as an instrumental environment for NLP system development and as an NLP learning environment. The learning environment systems can be divided into two main categories: learning tools and teaching tools (see Fig. 1).

TAISim toolkit can be attributed to the learning tools. Firstly, the system supports learning by self-contained invention, which means that the end-user can carry out experiments on text corpora and analyzes the results obtained after every step of NLP. Secondly, the toolkit enables learning by example: the end-user has an opportunity to choose information resources as well as software components used for different steps of analysis (grapheme analysis, morphological analysis, etc.) and to compare the results of processing of the same text obtained with the help of different resources. Thirdly, due to a meta-ontology that is not a domain ontology and describes a set of system's resources including software components, logging and a high-level description of the end-user's actions and related results, TAISim supports learning by explanation.

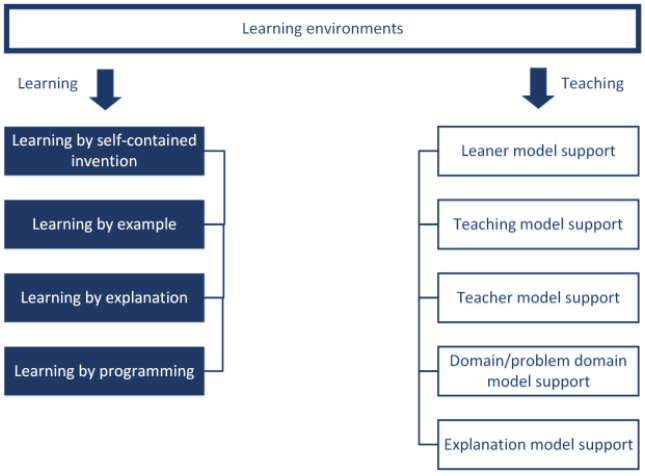

Fig. 1. Learning environments classification (adapted from [1])

Components that support learning by programming and implementation of self-developed units of NLP systems are under development. Now the end-user has an opportunity only to review the source code of different modules and has no opportunity to replace them with new ones created from scratch within TAISim.

After reengineering of the Serelex system, to use TAISim both as an NLP learning environment tools and as environment tools adaptable to automate the NLP systems development, first, we have designed a special high-level interface suitable to demonstrate the resources used and results obtained by separate steps of text processing consequentially. These steps are presented below in Fig. 2.

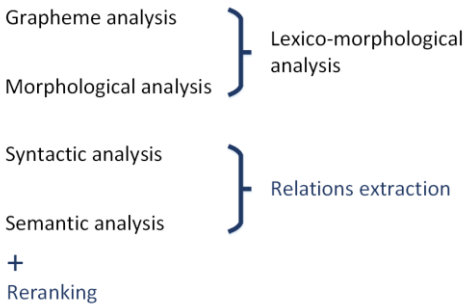

Fig. 2. NLP steps demonstrated within TAISim environment

TAISim interface is described in the next section of the paper.

3 Meta-tooling Framework TAISim

Let us consider the conception of the suggested approach for designing of a meta-tooling framework TAISim, which integrates an open source NLP components implementation of the Serelex as an essential part of the system with new visual components for text-based information retrieval and ontology learning methods exploration. The original corpus-based semantic similarity measure PatternSim, which was suggested by Alexader Panchenko [2], plays a key role in the Serelex lexico-semantic search engine and enables the system to retrieve terms semantically related to the query and rank them by a relevance. The measure provides results comparable to the baselines without the need for any fine-grained semantic resource such as WordNet [3].

It is known that the drawback of pattern-based approaches is of course the need to define the patterns, which is a time consuming but often very valuable task. Because TAISim has been built as a customizable system, it is possible within TAISim not only to demonstrate step by step different phases of text processing and compare the results of the lexico-semantic search engine by using different supporting tools and resources, but also to use a pattern-based approach to tackle problems related to the automation of an ontology extraction and refinement.

Fig. 3 shows fragments of the TAISim Environment Tools Suite interface with an example of “Ontology Summit 2014 Communique Big Data and Semantic Web Meet Applied Ontology” text processing [4].

We try to explore the applicability of the existing set of patterns for ontology extraction based not only on the text corpora, but also on the basis of the so-called “etalon” text. Then we integrate the extracted ontology into related concept hierarchy and establish conceptual relations from a set of external ontologies and thesauri, which are manually constructed or constructed with the help of ontology learning instruments from large text corpora. It is useful for a wide range of applications, in particular to automatically examine comprehensiveness of subject domain reviews or to automatically build so-called ‘ontology profiles’ with meta-data about every resource during its allocation into a repository in order to perform semantic indexing of documents.

For every pair of extracted concepts, a special re-ranking component adopted from the Serelex evaluates the similarity score [2]. The results of the concordance extraction are used both for evaluation of the semantic similarity between the concepts and for automatic ontology building. Fig. 4 presents a fragment of TAISim interface of the last two text processing steps before the visualization, which deals with re-ranking and converting the obtained results into JSON format.

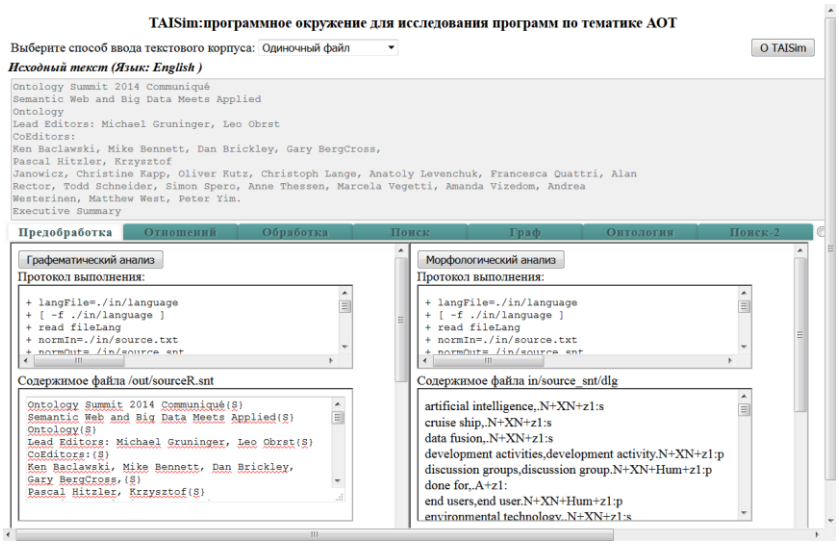

Fig. 3.1. A fragment of the TAISim Environment Tools Suite interface: grapheme and morphological analysis steps

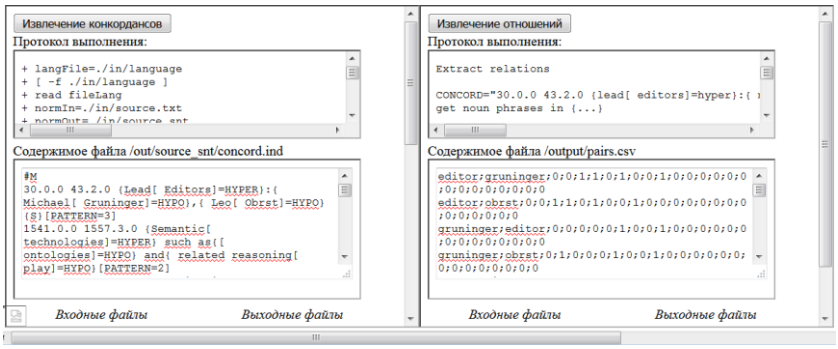

Fig. 3.2. A fragment of the TAISim Environment Tools Suite interface: concordances and relations extraction steps

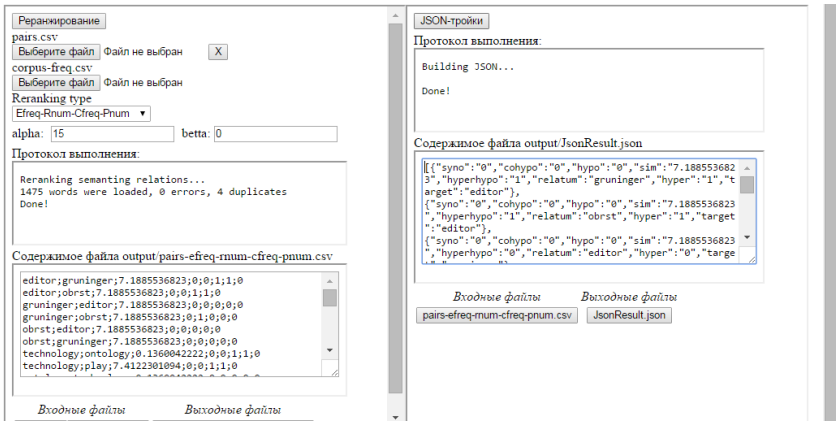

Fig. 4. Re-ranking and converting into *JSON* format

For greater clarity, we demonstrate a simple example of the establishment of relations of synonymy based on the processing of the following text fragment:

"The central problems (or goals) of AI research include reasoning, knowledge, planning, learning, natural language processing (communication), perception and the ability to move and manipulate objects".

For the beginning part of this input text fragment, the system builds the following concordance with the help of a lexico-syntactic pattern such as {NP=SYN} (or {NP=SYN}):

The {central[problems]=SYNO} (or{[goals]=SYNO})[PATTERN=11].

This concordance is used for building an ontology fragment representing the synonym relationship between the two concepts and then can be used for merging with the ontology base from the TAISim environment.

As can be seen from Fig. 5 due to visualization components not only the benefits of the pattern-based automation relation extraction but also the problems related to the necessity of collecting of context profiles per sense acquired from a training corpus to tackle word sense disambiguation problems have become more evident.

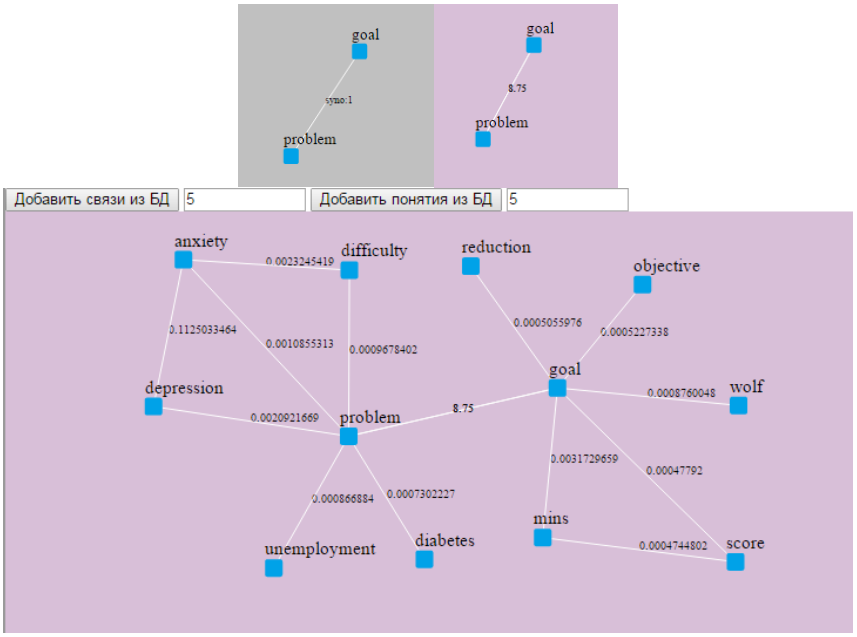

Fig. 5. An example of the relation extraction and enrichment

Different colors of work space are used to represent a different type of graphs: the grey color – to depict an ontology graph, and the light purple color – to depict a semantic similarity graph.

4 Architecture of Meta-tooling Framework TAISim

The architecture of meta-tooling framework TAISim is shown in Fig. 6. Within current TAISim prototype we use lexico-syntactic patterns both for the syntactic and semantic analyses. After that, a CSV file with the concordances extraction results is automatically created. The structure of this file is following:

$$< c_1, c_2, r_1, r_2, \dots, r_5, \text{sum}, p_1, p_2, \dots, p_{17} > \quad (1)$$

where c_1 and c_2 are terms; r_1 – the amount of "synonym" relations extracted; r_2 – the amount of "co-hyponym" relations extracted; r_3 – the amount of "hypernyms+hyponyms" relation extracted; r_4 – the amount of "hypernyms" relation extracted; r_5 – the amount of "hyponyms" relation extracted. The sum represents the total number of the patterns extracted the given pair of terms. The system supports 17 lexico-syntactic patterns [2] where p_i is the number of successful executions of i-pattern for the given pair of terms.

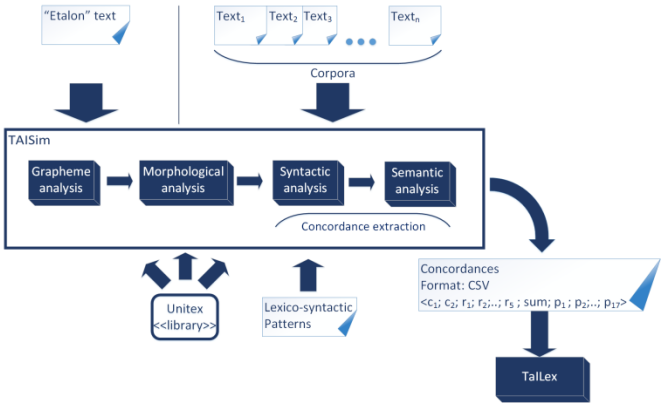

Fig. 6. Architecture of meta-tooling framework TAISim

The system allows the end-user to process texts in English or Russian. Russian texts processing has become possible thanks to lexico-syntactic patterns for the Russian language developed by A. Lukanin and K. Sabirova [5]. Fig. 7 demonstrates a part of ontology profile that has been extracted from the paper "Thesauri in information retrieval tasks" (author Loukachevitch N. [6]) by text processing within TAISim and its visualization with the help of TAILex components.

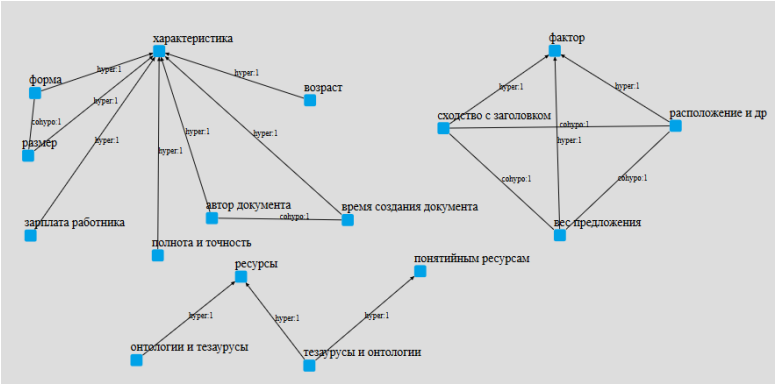

Fig. 7. A part of ontology profile has been extracted from [6]

PatternSim measure component uses Dela dictionary for morphological analysis. Our toolkit gives the alternative dictionary OpenCorpora as a part of pymorphy2 library in order to continue the NLP even if the dictionary has no information about a word. Cloud Content Repository C2R developed by IVS corporation in a collaboration with Small Innovative Enterprise (SIE) named KNOVA (one of the co-authors of the paper, S. Chuprina, is one of the co-founders of this SIE) is used as a document storage for the corpora and other types of information resources. For more detail, see [7, 8]. Now we are developing the special modules that allow not only creating a new

and enriching the existing TAISim information resources such as dictionaries but also enhancing, expanding and analyzing a set of lexico-syntactic patterns.

To upgrade TAISim with a new functionality that is intended to visualize the analytics results, a new component named TAILex has been implemented. The input of this component is the output of re-ranking process in CSV format enriched with the semantic relations extracted during the concordance construction process with the help of lexico-syntactical patterns [2, 3]. These data are converted to the JSON format and used as an input for TAILex subsystem to visualize the semantic relations between concepts extracted from "etalon" text or text corpus. Actually, it is a lightweight ontology visualization process. Besides that, TAILex provides the service to visualize the graph of semantic similarity obtained at the previous NLP steps and uses external linguistic resources such as Wikidata, ukWaC to enrich the obtained ontology with the semantically similar concepts and relationships.

Due to visualization TAILex not only helps the researchers or students to examine the extracted ontology profiles and to refine them but also it helps to find some drawbacks in the TAISim source code and to modify some lexico-syntactic patterns. The architecture of TAILex is shown in Fig.8.

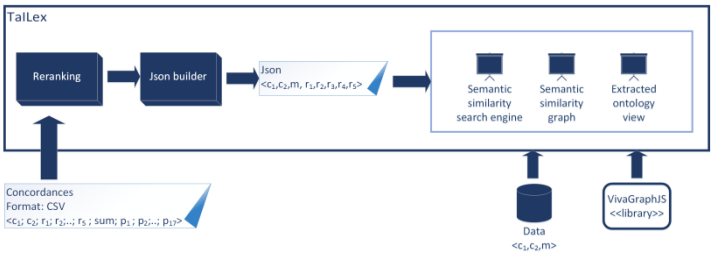

Fig. 8. Architecture of TAILex components

A registered end-user can look through current settings and follow the results through the log. This log will give the end-user an opportunity to analyze the results and collect the statistics about the effectiveness of usage of information and software resources.

5 Using Ontology-Driven Methods to Adapt NLP Systems to New Supporting Resources

If you design the framework that is aimed to automate the NLP system implementation, it is very important to achieve high level adaptability. An adaptable toolkit should provide not only an opportunity for a flexible configuration and extending the set of tools but also allows developing new tools to extend its own functionality within its own environment without any source code modification of the legacy system components. To complete this challenge, we use ontology-driven methods. Keeping in line with our approach to development of adaptable systems, ontologies are not

only the subject of study, but also the artifacts ready to be the basis for research tools development (see, for example, some of our previous projects under supervision of S. Chuprina, which is partially described in [7], [9,10,11]).

Within TASim project, we use ontology-engineering methods to construct an ontology named "system" ontology to accumulate a meta-knowledge about the system resources and program components, and about users' activities to control the process of development and using the NLP system. Thanks to ontology-driven methods TAISim can be modified and enriched with additional processing and information resources by means of a high-level user interface. Fig.9 depicts an example of such "system" ontology. From Fig. 9 you can see that the functions of UNITEX corpus processing system are used during the grapheme, morphological and syntactic analysis steps. The first one consists of four sub-steps (Extra separators removal, Sentence splitting, Short form expanding, Tokenization), which use UNITEX functions (Normalize, Fst2Txt, Tokenize) and hand-crafted graphs in GRF format (Sentence.grf and Replace.grf) as supporting resources.

You can see also that some steps, for example, Sentence splitting and Short form expanding, require only one type of supporting resources, and others, for example, step of morphological analysis, use both Dela dictionaries and Unitex function Dico. The end-user has an opportunity to choose one dictionary (for example, Dela or DelaNew) or to combine them as well. Fig. 9 demonstrates only a high-level ontology that can be viewed by a casual user. But the developers can extract and modify also a task ontology, which represents a more deeper layer of knowledge (see Fig. 10). The syntactic analysis step uses the Unitex Concord/Locale functions and Lexico-syntactic patterns and then the results of this step in a form of extracted concordance will be processed by the semantic analysis procedure, which is the same one as in Serelex. At the last step the semantic analysis results are ranking according to a chosen re-ranking type.

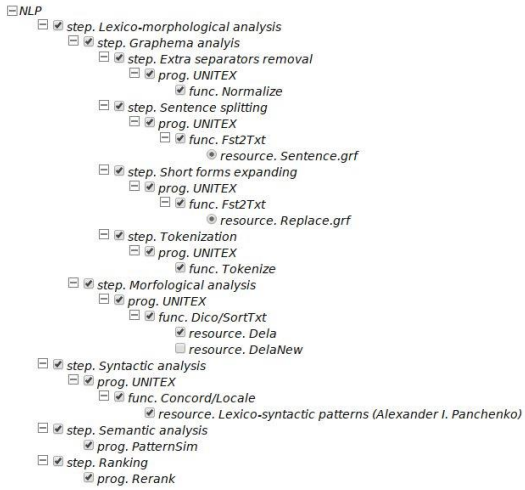

Fig. 9. A part of ontology with meta-knowledge about TAISim NLP components and resources

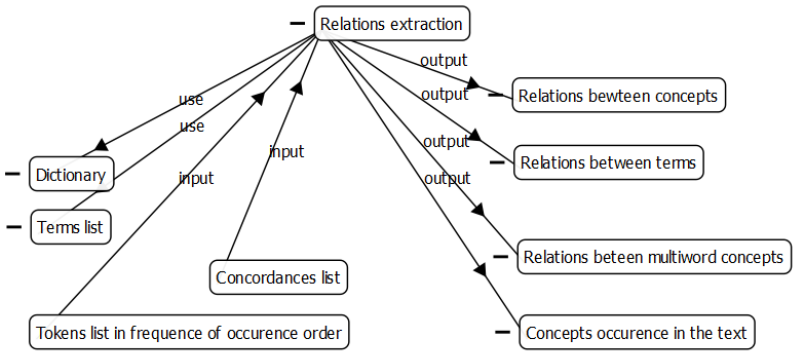

Fig. 10. A part of task ontology

6 Conclusion

The paper is devoted to description of an approach for NLP system development using ontology-driven methods. We have demonstrated that TAISim meta-tooling framework can help the end-user to develop and perform NLP system based on PatternSim method and to study each step results. The framework can be used both as a developer's tool for creating NLP systems and as a NLP learning environment. Under the proposed approach, the end-user can change the system's components with new ones without any changes in the source code of other legacy components due to special control mechanism based on the meta-ontology ("system" ontology), which describes the meta-knowledge about the TAISin construction.

Now, TAISim is at the stage of a User Experience Prototype and the demo prototype of TAISim without an opportunity to add new processing sources and patterns is free accessible on the site http://gate.psu.ru:45080. As mentioned above, due to new TAILex visualization components it has become possible to automate the ontology construction that helps to tackle a wide range of semantic search problems and other ones, for example, Machine Translation problems (see [12, 13]). In future work, we are going to extract a set of patterns and to study the usefulness of some Ontology Design Patterns from http://OntologyDesignPatterns.org to improve the quality of semantic relations extraction for automatic ontology profiles construction.

To demonstrate the practical viability of the proposed approach, we use ontology profiles, which are extracted from the bilingual text corpus in Computing Area, as the initial source for ontology construction within a real life project centered on "Development Models and Tools to Transform Traditional Information Systems into Intelligent Systems via the use of a Bilingual Ontology in the Computing Area" with the help of the adaptable ontology editor ONTOLIS[5] [7], [10]. We also have done a series

[5] The reported study was partially supported by the Government of Perm Krai, research project No. C-261004.08

of successful experiments using TAISim to generate ontology profiles for semantic indexing of different types of information resources within C2R [7, 8].

References

1. Lyaschenko, N.: Analysis of computer-based learning system models. Submodeling for advanced training computer systems. Fundamental Research Scientific Journal, 10, 2153-2157, http://www.fundamental-research.ru/ru/article/view?id=32726 (In Russian)
2. Panchencko, A.: Similarity Measures for Semantic Relation Extraction. PhD thesis. Universite catholique de Louvain, Louvain-la-Nauve, Belgium, 196, http://cental.fltr.ucl.ac.be/team/panchenko/thesis.pdf
3. Panchenko, A., Morozova, O., Naets H.: A Semantic Similarity Measure Based on Lexico-Syntactic Patterns. In: Proceedings of the 11th Conference on Natural Language Processing (KONVENS 2012), pp. 174-178, Vienna (2012), http://www.oegai.at/konvens2012/proceedings/23_panchenko12p/
4. Ontology Summit 2014 Communique Big Data and Semantic Web Meet Applied Ontology, http://www.inf.unibz.it/~okutz/resources/OntologySummit2014_Communique_v1-0-0_20140429-1045.pdf
5. Sabirova, K., Lukanin, A.: Automatic Extraction of Hypernyms and Hyponyms from Russian Texts. In: Supplementary Proceedings of the 3rd International Conference on Analysis of Images, Social Networks and Texts (AIST'2014), vol. 1197, pp. 35-40. CEUR-WS, Yekaterinburg (2014)
6. Loukachevitch, N.: Thesauri in information retrieval tasks (2010), http://www.nsu.ru/xmlui/bitstream/handle/nsu/9086/louk_book.pdf
7. Chuprina, S.: Steps Towards Bridging the HPC and Computational Science Talent Gap Based on Ontology Engineering Methods. In: Procedia Computer Science, vol. 51, pp. 1705- 1713. Elsevier (2015)
8. Kostarev, A.: Cloud Content Repository C2R: Text Analysis Based on Ontology Engineering Methods. In: Proceedings of the 9th Conference "Free Software in High School", pp. 23-28 (2014) (In Russian)
9. Ryabinin, K., Chuprina, S.: Development of ontology-based multiplatform adaptive scientific visualization system. Journal of Computational Science, vol. 10, pp. 370-381. Elsevier (2015)
10. Chuprina, S. Zinenko, D.: Adaptable Visual Ontological Editor ONTOLIS. Vestnik of Perm State University. Math. Mechanic. Information science, 3, 106-110 (2013) (In Russian)
11. Chuprina, S., Nikiforov, V.: Using Ontological Engineering Methods to Improve Graphical Resources Indexing for Sensible Searching. Vestnik of Perm State University. Math. Mechanic. Information Science, 3, 113-118(2013) (In Russian)
12. Kostareva, T., Chuprina, S.: Conception of ontology-driven designing of NLP systems. Vestnik of Perm State University. Math. Mechanic. Information Science, 5, 108-114 (2015) (In Russian)
13. Knight, K.: Building a large ontology for machine translation. In: Proceeding HLT `93 Proceedings of the workshop on Human Language Technology, pp. 185-190, Stroudsburg, PA, USA (1993), http://www.aclweb.org/anthology/H93-1036