Computational Linguistics and Intellectual Technologies:
Proceedings of the International Conference "Dialogue 2017"

Moscow, May 31—June 3, 2017

SEMANTIC DESCRIPTIONS FOR A TEXT UNDERSTANDING SYSTEM

Boguslavsky I. (bogus@iitp.ru)

Laboratory of Computational Linguistics, A. A. Kharkevich Institute for Information Transmission Problems, Russian Academy of Sciences, Russia

Departamento de Inteligencia Artificial, ETSII, Universidad Politécnica de Madrid, Spain

The semantic analyser SemETAP is a module of the ETAP-3 Linguistic Processor. It uses 2 static semantic resources—the combinatorial dictionary and the ontology. The former contains multifarious information about the words, and the latter stores extralinguistic (world) knowledge on the concepts and serves as the metalanguage for semantic description. World knowledge is needed, on the one hand, to enhance text analysis, and, on the other hand, to extract implicit information by means of inference. Both words and concepts are supplied with semantic descriptions. A semantic description consists of a definition in a formal language, which can optionally contain implications and expectations. For user's convenience, the description may also be provided by examples and a definition in NL. Semantic descriptions of several words and concepts are given.

Keywords: language model, ontology, deep text analysis, semantic definitions, implications, expectations

1. SemETAP Semantic Analyser

Semantic analyser, called SemETAP, is a module of the ETAP-3 multifunctional linguistic processor. Its goal is to provide semantic interpretation of texts using linguistic and world knowledge. Examples of how SemETAP operates can be found in Boguslavsky et al. 2015 and Boguslavsky 2016. Semantic descriptions of words and concepts are a key component of SemETAP. The content and the format of these descriptions are determined by the design and the goals of SemETAP. Its main features are as follows.

- Rule-based approach. The system is mostly knowledge-based, although some modules contain data-driven components.
- Stratification. Each sentence is represented by a series of structures, which correspond to various representation levels. These are (a) Morphological structure, (b) Syntactic structure, (c) Normalised syntactic structure, (d) Basic semantic structure, and (e) Enhanced semantic structure.
- Balance between the rules and the dictionary. Linguistic knowledge is distributed among two resource types—static (dictionaries and ontology) and dynamic (sets of rules), which interact strongly.
- Linguistic and world knowledge. As opposed to many other semantic processing systems, including advanced semantic parsers, such as StanfordCoreNLP[1], Boxer[2], WASP[3] and KRISP[4], SemETAP uses not only linguistic but also world knowledge. We share this approach with several other knowledge-oriented projects, which also rely on using detailed semantic and ontological information (cf. e.g. Pustejovsky 1991, Nirenburg and Raskin 2004, Mairal and Usón 2009, Anisimovich et al. 2012, Cimiano et al. 2014). Semantic descriptions created for SemETAP are distributed between the Combinatorial dictionary and the Ontology. Both resources use the same metalanguage, based on the ontological elements.
- Focus on inference. We assume that the level of text understanding achievable by the semantic analyser is determined by the amount of inferences the system can draw. Therefore, the major goal of the analyser is twofold: it should (a) construct the Basic Semantic Structure (BSemS) of each sentence, and (b) draw all possible inferences from BSemS, which results in the Enhanced Semantic Structure (EnSemS).

Among the inferences envisaged in semantic descriptions, we distinguish between strict logical entailments (implications) and plausible expectations. Both play an important role in interpreting coherent discourse and dialogues. An implication of an utterance is an inference that is necessarily true. For example, sentence *John broke the cup* necessarily implies that the cup has lost its integrity. A plausible expectation takes place when a certain state-of-affairs can be expected in the given situation but it is not obligatory. When somebody says that *John dropped the cup* we can reasonably expect that the cup will be broken, but we will easily accept the opposite. Sometimes, an utterance allows for both types of inference. For example, the literal meaning of the sentence *John went to the university (at moment t)* that constitutes its BSemS is that at *t* John began moving towards the university with the aim of being there. Out of this BSemS, one can draw two conclusions that differ in power. The first one has the status of a logical entailment and hence is completely true: "at *t* John ceased to be at the initial point of his movement". The second inference is merely a plausible expectation: "it can be expected that at some later moment *t1* John will be at the university".

---

1 http://nlp.stanford.edu:8080/corenlp/

2 http://www.let.rug.nl/bos/pubs/Bos2015NoDaLiDa.pdf

3 http://www.cs.utexas.edu/~ml/wasp/

4 http://www.cs.utexas.edu/~ml/krisp/

It can be shown that during the interpretation of discourse, plausible expectations play an even greater role than logical implications. Apparently, it is plausible expectations that mostly guarantee text coherence and help restore omitted pieces of information. For example, the sentence *Mother asked me to repair the fence* does not logically imply that the fence has been repaired. However, this is a plausible expectation entailed by the meaning of *asked*. Therefore, we perceive the following dialogue as coherent: *What were you doing yesterday?—Mother asked me to repair the fence.* Although the reply does not give a direct answer to the question, the hearer extracts the answer due to the expectation generated by *asked*. In compiling semantic descriptions, we tried to pay special attention both to implications and to plausible expectations.

2. Ontology and Combinatorial dictionary

As mentioned above, the system disposes of two semantic resources—the Combinatorial dictionary and the Ontology. The Ontology plays a double role in the project. On the one hand, it serves as a structured source of world knowledge; on the other hand, ontological elements (concepts, instances, ontological relations) constitute a unique metalanguage of semantic description. This means that all sense-bearing text elements should be interpreted in ontological terms. This makes the task of establishing the links between the dictionary and the ontology far from trivial (Boguslavsky et al. 2010).

Combinatorial dictionary has a ramified structure and contains many types of information (Apresian et al. 2003). It is distributed among the following zones:

1 Lexeme name
2 Syntactic features
3 Semantic features
4 Government pattern (subcategorization frame)
5 Lexical functions
6 Zones of translation to another language—a separate zone for each working language (English, German, Spanish, Korean, UNL, Ontology-based semantic language)[5]
    6.1 Default equivalent
    6.2 Translation rules
7 Other types of rules operating at various stages of processing.

Dictionary entry zones relevant for this paper are the Semantic features zone (3), Government pattern (4) and the Semantic language zone (6).

Semantic features are used in all ETAP-3 options, including semantic analysis. They are referred to by the rules of various types, first of all by semantic agreement rules. Until recently we used a set of 57 features. Last year these features were replaced by ontology concepts, which allow a much more detailed representation of semantic

[5] ETAP-3 is a multilingual system, and the dictionary is designed so as to permit each word to be translated to several languages. For each translation language, the dictionary entry has a special zone. For this paper, only translation to the semantic language is relevant.

properties of words. One of the consequences of this operation is that semantic restrictions in the government pattern can now be formulated in terms of ontological concepts and not in terms of semantic features used before.

If the word has valencies, its syntactic government pattern is supplemented by the semantic one, to show which element of BSemS corresponds to each syntactic actant.

Zone 6 of the entry gives a semantic equivalent of the word. If the word has a direct correspondence among the ontology concepts, it is given in zone 6.1 "Default equivalent". The semantics of this concept is described in the ontology. If the ontology contains no direct equivalent of the word, and its introduction is not expedient for any reason, semantic description of the word is given in zone 6.2 by means of a full-fledged BSemS. The choice between these alternatives depends on various considerations on which we cannot dwell here. Still, this choice is often a matter of convenience, because both the words and the concepts are described in the same metalanguage and according to the same principles.

OntoEtap ontology is built on the basis of the popular and freely downloadable SUMO ontology (www.ontologyportal.org), which we supplemented by a considerable amount of data necessary for semantic analysis of natural languages (Boguslavsky 2011). An important property of all ontologies, which we actively exploit, is the top-down inheritance of all properties.

From the formal point of view, a semantic structure is a set of triples of the type `relation(Ontoelement-1,Ontoelement-2)`, where `relation` is an object or data property of the ontology, and `Ontoelement-i` is a variable or a constant denoting a concept or an instance. This rdf-formalism was chosen because, on the one hand, it is very flexible and expressive, and on the other hand, it is supported by a wide range of tools and is easily integrated with many Semantic Web applications.

3. Semantic descriptions of words and concepts

As mentioned above, semantic descriptions of words and concepts are carried out in the same semantic metalanguage and according to the same principles. As of now, we described a number of concepts and Russian words belonging to various semantic classes[6]: mental predicates (*want, patience, understand*), events (*ball, examination, interview*), instruments (*saw, axe, frying-pan*), animals (*dog*), plants and fruit (*apple, olive*), body parts (*hand, face, breast, pelvis*), time (*noon, midnight, soon*), subjective attributes (*cautious, dangerous, sympathetic, daring*), natural phenomena (*frost, hot weather, cool weather*), emotional states (*anger, grieve, resentment*), transport (*airplane, helicopter*), organizations (*restaurant, library*), and some other.

In completeness and detail, semantic descriptions are close to modern lexicographic definitions, but often surpass them in the amount of world knowledge. However, these descriptions do not replace encyclopedia. We include only such world knowledge that may be useful for commonsense reasoning—although clear boundaries are obviously very difficult to draw.

---

[6] For simplicity, we represent examples with English words rather than concept names or Russian words.

As an illustration, we provide a layout for the description of physical objects. In parentheses, we give corresponding ontological relations. Aspects to be taken into account while describing an Object include the following:

- parts of the Object; obligatory (hasPart): *bird—wing, house—roof;* or typical (hasTypicalPart): *house—attic, loft, cellar.*
- something that the Object is part of (inverse to hasPart and to hasTypicalPart—isPartOf, isTypicalPartOf): *window—building, transport*.
- typical size (height, weight …) of the Object (hasSize, hasHeight, hasWeight, …): *apple—10 cm*.
- typical material or an object the Object consists or is made of (isMadeOf): *book—paper, book cover—cardboard, fruit juice—fruit, porc—pig*.
- things that are typically made of the Object (inverse to isMadeOf—isMaterialFor): *fruit—juice, milk—cheese, timber—furniture, wood—furniture, gold—jewelry*.
- typical form of the Object (hasForm): *pill—round*.
- typical colour of the Object (hasColour): *apple—red, green, yellow.*
- typical location of the Object (hasTypicalLocation): *fish—in a natural body of water* or *in an aquarium, fruit—in the orchard, cloudberry—in the tundra*.
- typical origin (hasOrigin): *avocado—Southern region, camembert—France*.
- major predestination of the Object (hasFunction): *axe—chop* (as an instrument), *pen—write* (as an instrument), *food—eat* (as an object), *beverage—drink* (as an object). The predestination of *hen* for being eaten is accounted for by the fact that it is included not only in the class Poultry (which does not have any predestination) but also in the class Food (which does) and inherits hasFunction Eating (as an object) from this class.
- situations in which the Object frequently takes part, different from the main predestination (participatesIn): *axe—draw nails* (as an instrument), *knife—kill* (as an instrument), *hen—boil, fry, feed* (as an object), *lay eggs* (as the subject).

Some of these data, such as the typical location or frequent situations are placed in the section Expectations (see below), if the probability of their being true in all cases is not high enough.

In a general case, a semantic description contains the following sections:

1. Examples.
2. Definition or explanation in natural language.
3. Definition in a formal language, which may include Implications and Expectations.

The first two sections are intended for humans and written in natural language, and the third section is written in the formal language and used for semantic analysis and inference. From the formal point of view, the definition is a rule whose left part is the word (or a concept) and possibly a set of conditions, and the right part is a BSemS.

Below are several semantic descriptions of words and concepts of different classes supplied with detailed comments.

4. Examples of semantic descriptions

Below we will illustrate semantic descriptions by one word (*pomogat'* 'to help') and several concepts.

4.1. *Pomogat'* 'to help'

We will take the word *pomogat'* 'to help' in its major sense represented in examples (1)–(4). In square brackets are elements of BSemS (which will be explained below) corresponding to the actants of *pomogat'*.

<u>Examples:</u>

(1) *Kolja* [Agent1] *pomogaet Mashe* [Agent2] *reshat'* [Event2] *zadachu*.
'Kolja [Agent1] helps Masha [Agent2] solve [Event2] the problem'

(2) *Uchitel'* [Agent1] *pomogaet ucheniku* [Agent2] *v vybore* [Event2] *temy dlja sochinenija*.
'the teacher [Agent1] helps the pupil [Agent2] to chose (lit. in the-choice of) [Event2] the topic for the composition'

(3) *On* [Agent1] *pomog mne* [Agent2] *s perevodom v Moskvu* [Event2] *i s zhiljem* [Object1].
'he [Agent1] helped me [Agent2] with the transfer to Moscow [Event2] and with the lodging [Object1]'

(4) *On* [Agent1] *vsegda gotov pomoch den'gami* [Object2] *i sovetom* [Event4].
'he [Agent1] is always willing to help with money [Object2] and advice [Event4]'.

<u>NL definition</u>:
"Agent1 has the goal of doing Event2 or obtaining Object1. Agent2 has the goal of facilitating this to Agent1. Therefore Agent2 is doing Event4 or is giving Object2 to Agent1. It is good for Agent1 that Agent2 is doing this".

<u>Formal definition</u>:
To make the formal definition more illustrative, we will represent it by a commented table.

Pomogat' →	
hasObject(?Goal1, ?Agent1)	?Agent1 has the goal of performing ?Event1,
hasObject2(?Goal1, ?Event1)	
hasAgent(?Event1,?Agent1)	
hasAlternative(?Event1, ?Event2)	which is either ?Event2 (*solve* in (1), *chose* in (2), *transfer* in (3))
hasAlternative(?Event1,?Getting)	or getting
hasObject(?Getting,?Object1)	?Object1 (*lodging* in (3))
hasObject(?Goal2,?Agent2)	?Agent2 has the goal of
hasObject2(?Goal2,?Facilitating)	facilitating

hasObject(?Facilitating,?Event1)	?Event1
hasBeneficiary(?Facilitating,?Agent1)	for ?Agent1
hasAgent(?Event3,?Agent2)	?Agent2 performs ?Event3
hasAlternative(?Event3,?Event4)	which is either ?Event4 (*advice* in (4))
hasAlternative(?Event3,?Giving)	or giving
hasObject(?Giving,?Object2)	?Object2
hasRecipient(?Giving,?Agent1)	to ?Agent1
hasObject(?EvalModality, ?Event3)	?Event3 is good
hasBeneficiary(?EvalModality, ?Agent1)	for ?Agent1
hasValue(?EvalModality, HighDegree)	

Implication: if *POMOGAT′* = past,perf, then Agent2 performed Event3 and Agent1 performed Event1.

Petr pomog Mashe reshit' zadachu → *Masha reshila zadachu*
'Petr helped Masha solve the problem' → 'Masha solved the problem'

Uchitel' pomog ucheniku v vybore temy → *Uchenik vybral temu*
'The teacher helped the pupil choose the topic' → 'The pupil chose the topic'

On pomog mne s perevodom v Moskvu → *Ja perevelsja v Moskvu*
'He helped me with the transfer to Moscow' → 'I transferred to Moscow'

On pomog mne s zhiljem → *Ja poluchil zhilje*
'He helped me with the lodging' → 'I got the lodging'

On pomog mne den'gami i sovetom → *On dal mne den'gi i sovet, i ja sdelal to, chto xotel sdelat'*
'He helped me with money and advice' → 'He gave me money and advice, and I did what I wanted to'

Note the last example: although the initial sentence does not mention the goal that Agent2 wishes to achieve, one can infer that the goal has been met.

Expectation: if *POMOGAT′* =nonpast or imperf, then it can be expected that: Agent2 performs Event3 and Agent1 performs Event1.

Petr pomogaet (pomozhet) Mashe reshit' zadachu → It can be expected that: *Masha reshit zadachu*.
'Petr helps (will help) Masha solve the problem' → It can be expected that: 'Masha will solve the problem'

Below are descriptions of concepts.

4.2. Apple

Example:

Eva sorvala s dereva jabloko i ugostila Adama
'Eva plucked an apple and gave it to Adam'

NL definition: "A fruit as big as a fist growing on apple tree, of round shape. Having a red, yellow or green colour, contains juicy flesh, peel, small brown seeds, good for health, of sweet or sour-sweet taste".

Formal definition:

Apple(?Apple) →	If there is an instance ?Apple of the Apple concept, then:
Fruit(?Apple)	it belongs to the Fruit class. The latter, in its turn, belongs to the Food class, whose predestination is being eaten. The description of Eating includes the proposition that the goal of eating is to satisfy hunger or to enjoy. All these data are inherited by Apple and other Fruit.
hasObject(?BeFruitOf, ?Apple)	Apple is a fruit of an apple tree.
hasObject2(?BeFriutOf, ?AppleTree)	
hasPart(?Apple, ?Thing1)	Here major parts of an apple are listed: - seeds, which are: small brown - stem - skin - juice
hasSubset (?Thing1, ?Seed)	
hasSize(?Seed, Small)	
hasColor(?Seed,Brown)	
hasSubset (?Thing1, ?Stem)	
hasSubset (?Thing1, ?Skin)	
hasSubset (?Thing1, ?Juice)	
hasObject(?HavingSize,?Apple)	Here the size of a typical apple is given. Since our descriptions are intended for commonsense reasoning, we prefer to describe the size of objects in absolute numbers, though approximate, and not by means of anthropomorphic reference ("size of a fist"), as it is done in lexicography. The typical size of an apple is about 10 centimetres.
hasValue(?HavingSize, ?LinearMeasure)	
inUnit(?LinearMeasure, Centimeter)	
hasNumericalValue(?LinearMeasure, 10)	
has Attribute(?Apple,?Attribute)	Apple has several attributes:
hasSubset(?Attribute, ?ColorAttribute)	- colour,
hasSetOrAlternative(?ColorAttribute, Red)	which can be red
hasSetOrAlternative(?ColorAttribute, Yellow)	yellow or
hasSetOrAlternative(?ColorAttribute, Green)	green hasSetOrAlternative relation denotes non-exclusive disjunction, as opposed to hasAlternative, which corresponds to exclusive disjunction

hasSubset(?Attribute, Round)	- round shape
hasSubset(?Attribute,?TasteAttribute)	- taste, which can be
hasAlternative(?TasteAttribute, Sweet)	sweet or
hasAlternative(?TasteAttribute, Sour-sweet)	sour-sweet
hasSubset(?Attribute, GoodForHealth)	- good for health
hasSubset(?Attribute, Juicy)	- juicy
hasSubset(?Attribute, Crisp)	- crisp

Expectations:

participatesIn(?Apple,?Eating)	Typical situations in which apples participate:
hasObject(?Eating,?Apple)	- Eating (as an object) (inherited from Food)
participatesIn(?Apple, ?Baking)	- Baking (as an object)
hasObject(?Baking, ?Apple)	
participatesIn(?Apple, ?Squeezing)	- squeezing apple juice
hasObject(?Squeezing, ?Apple)	
hasResult(?Squeezing, ?AppleJuice)	
participatesIn (?Apple, ?Making)	- Making such objects as:
hasObject(?Making, ?Thing2)	
hasSubset(?Thing2, ?ApplePie)	ApplePie
hasSubset (?Thing2, ?AppleJam)	AppleJam
hasSubset (?Thing2, ?Cider)	Cider
isMaterialFor(?Apple, ?Thing2)	Apple participates in the manufacturing of these objects as an ingredient
hasTypicalLocationAt(?Apple, ?Thing3)	Typical places where one can find Apple:
hasAlternative(?Thing3, ?Orchard)	- Orchard
hasAlternative (?Thing3, ?AppleTree)	- AppleTree
hasAlternative (?Thing3, ?GroceryStore)	- GroceryStore
hasAlternative (?Thing3, ?House)	- House
hasAlternative(?Thing3, ?Bowl)	- Bowl
hasAlternative (?Thing3, ?Fridge)	- Fridge

4.3. Heating

Example: *Nagrevaem smes' do kipenija, a potom oxlazhdaem* 'we heat the mixture until it boils and then cool it'. *Prodavcy tropicheskix rybok obogrevali akvariumy kero-sinovymi lampami* 'the sellers of tropical fish heated aquariums with oil lamps'.

NL definition: "The temperature of ?Object increases from ?Quant1 to ?Quant2"

Formal definition:

Heating(?Heating) →	if there is an instance ?Heating of Heating, then:
IncreasingProcess(?Heating)	it belongs to the IncreasingProcess class
hasObject(?Heating, ?Object)	there is an Object that undergoes this process
hasTime(?Heating, ?TimeInterval)	over the time interval ?TimeInterval
begins(?Time1, ?TimeInterval)	?Time1 is the beginning of ?TimeInterval
ends(?Time2, ?TimeInterval)	?Time2 is the end of ?TimeInterval
hasObject(?HavingTemperature1, ?Object)	at ?Time1 ?Object's temperature is equal to ?Quant1
hasValue(?HavingTemperature1, ?Quant1)	
TemperatureMeasure(?Quant1)	
hasTime(?HavingTemperature1, ?Time1)	
hasObject(?HavingTemperature2, ?Object)	at ?Time2 ?Object's temperature is equal to ?Quant2
hasValue(?HavingTemperature2, ?Quant2)	
TemperatureMeasure(?Quant2)	
hasTime(?HavingTemperature2, ?Time2)	
greaterThan(?Quant2, ?Quant1)	?Quant2 is greater than ?Quant1

4.4. HeatingDevice

Example: *Nagrevatel'noe ustrojstvo USP-2 prednaznacheno dlja podogreva plastin na raznyx stadijax analiza* 'the heating device USP-2 is intended for heating plates at various stages of the analysis'.

NL definition: "A device that serves as an instrument of heating something, e.g. electric heaters, heat lamps, ovens, stoves, etc."

Formal definition:

HeatingDevice(?HeatingDevice) →	if there is an instance ?HeatingDevice of HeatingDevice, then:
Device(?HeatingDevice)	?HeatingDevice belongs to the ?Device class
hasFunction(?HeatingDevice, ?Heating)	The function of ?HeatingDevice consists in serving as an instrument in the ?Heating process
hasInstrument(?Heating, ?HeatingDevice)	

4.5. Stove

Example: *Nekotorye pechi rabotajut na neetilirovannom benzine* 'some stoves are fuelled with unleaded petrol'.

NL definition: "A device used for heating a room or for cooking, which works by burning wood, coal, oil, petrol or gas or is powered by electricity".

Formal definition:

Stove(?Stove) →	If there is an instance ?Stove of the Stove concept, then:
HeatingDevice(?Stove)	it belongs to the HeatingDevice class
hasFunction(?Stove, ?Heating1)	serving for heating is inherited from HeatingDevice (cf. above)
hasInstrument(?Heating1, ?Stove)	
hasGoal(?Heating1, ?Event1)	in the Stove, the heating is made either for
hasSubsetOrAlternative(?Event1, ?Heating2)	heating buildings or parts thereof (StationaryArtifact)
hasObject(?Heating2, ?StationaryArtifact)	
hasSubsetOrAlternative(?Event1, ?Cooking)	or for cooking, or for both
isResultOf(?Heating, ?Event2)	Heating is obtained either by
hasAlternative(?Event2,?Burning)	- burning
hasObject(?Burning,?Substance)	wood or
hasAlternative(?Substance, ?Wood)	coal or
hasAlternative(?Substance,?Coal)	oil or
hasAlternative(?Substance,?Oil)	petrol or
hasAlternative(?Substance,?Petrol)	gas
hasAlternative(?Substance,?Gas)	
hasAlternative(?Event2,?Using)	- or by using electricity
hasObject(?Using,?Electricity)	

4.6. Organization

Example: *Many international organizations have their headquarters in Geneva.*

NL definition: "Group of people whose activity is coordinated to attain common goals".

Formal definition:

Organization(?Organization) →	
Group(?Organization)	Organization belongs to two classes—Group
Agent(?Organization)	and Agent
hasChief(?Organization,?Human1)	Organization has a chief
hasInStaff(?Organization,?Human2)	and staff
hasFunction(?Organization,?Action)	Organization has a primary function—to do something

4.7. ClientServingOrganization

NL definition: “Organization whose function is to provide services to clients”.

Formal definition:

ClientServingOrganization (?CS-Organization) →	
Organization(?CS-Organization)	CS-Organization belongs to Organization and inherits all its properties
hasUser(?CS-Organization,?Agent)	CS-Organization has users that may be people or organizations
hasSubset(?Agent,?Human)	
hasSubset(?Agent, ?Organization)	
hasUserAction(?CS-Organization,?Action)	there is a typical action that a user of the CS-Organization performs. E.g. in a shop it is buying things, in a hospital it is receiving treatment and in a movie theatre it is watching a film.

4.8. Library

NL definition: “Organization that has a collection of sources of information and similar resources, and makes them accessible to clients”

Formal definition:

Library(?Library)	
ClientServingOrganization(?Library)	Library is a subclass of ClientServingOrganization and inherits all its properties (which we do not repeat here)
belongsTo(?Library, ?PhysicalObject1)	Library belongs to Organization or Region. The belongsTo slot is often filled in Russian by adjectives or genitive noun phrases: *Rajonnaja* ‘regional’, *gorodskaja* ‘city’, *shkol’naja* ‘school’, *sinodal’naja* ‘synodal’, *tjuremnaja* ‘prison’, *Administracii prezidenta* ‘President’s Administration’, *Akademii nauk* ‘Academy of Sciences’, *zavodskaja* ‘factory’, *oblastnaja* ‘provincial’, *kraevaja* ‘territorial’, *kafedral’naja* ‘departmental’, *korolevskaja* ‘royal’, *nacional’naja* ‘national’, *polkovaja* ‘regiment’, *universitetskaja* ‘university’
hasSubset(?PhysicalObject1, ?Organization)	
hasSubset(?PhysicalObject1, ?Region)	

hasUser(?Library, ?Human)	Users of libraries are also often expressed in Russian by adjectives: *kursantskaja* 'for cadets', *oficerskaja* 'for officers', *detskaja* 'for children', *obschedostupnaja* 'public', *rabochaja* 'for workers'
hasFunction(?Library,?Lending)	The function of the library consists in lending ContentBearingObjects to its users.
hasObject(?Lending1,?ContentBearing Object)	
hasAddressee(?Lending1,?Human)	
hasInStock(?Library,?ContentBearing Object)	Library disposes of ContentBearingObjects of different kinds: books, journals, newspapers, audios, videos, maps, patents, etc.
hasUserAction(?Library,?Intentional PsychologicalProcess)	What users are doing is reading listening to or watching these ContentBearingObjects
hasSubsetOrAlternative(?Intentional PsychologicalProcess,?Reading)	
hasSubsetOrAlternative(?Intentional PsychologicalProcess,?Listening)	
hasSubsetOrAlternative(?Intentional PsychologicalProcess,?Watching)	
hasAgent(?IntentionalPsychological Process, Human)	
hasObject(?IntentionalPsychological Process,?ContentBearingObject)	
hasLocation(?Library,?PhysicalObject2)	?Library may be located in an ?Organization or in a ?Region: *moskovskaja* 'Moscow', *domashnjaja* 'home'
hasSubset(?PhysicalObject2, ?Organization)	
hasSubset(?PhysicalObject2,?Region)	
hasAboutness(?Library,?Entity)	?Library may cover a definite topic or domain, which is often expressed by adjectives: *istoricheskaja* 'historical', *medicinskaja* 'medical', *muzykal'naja* 'musical', *pedagogicheskaja* 'pedagogical', *politexnicheskaja* 'politechnical', *spravochnaja* 'reference', *teatral'naja* 'theater', *po obschestvennym naukam* 'social sciences', *estestvennyx nauk* 'natural science', *nauchnoj fantastiki* 'science fiction'.
hasTypicalPart(?Library, ?ReadingHall)	Library often has ReadingHall

Implications:

- hasAboutness(?Library,?Z) → hasInStock(?Library,?Z1)&hasAboutness(?Z1,?Z)) (If Library covers subject domain Z, then ContentBearingObjects it contains have topic Z, i.e. a historic library contains books on history)
- hasLocation(?Library,?Z)&Organization(?Z) → hasUser(?Library,?Z1)&(hasInStaff(?Z,?Z1)/hasUser(?Z,?Z1)) (If Library is located in Organization, its users are either clients of this Organization or its employees, i.e. users of a university library are either students or university employees)

5. Conclusion

Our approach to semantic analysis lies within the knowledge-based paradigm. We are guided by the conviction that using explicit and detailed knowledge on the language and on the subject domain can be beneficial for many tasks. The compilation of detailed semantic descriptions, which include both linguistic and extralinguistic knowledge, is important in different perspectives. On the one hand, they are needed for modeling language competence, in the direction of both understanding and generation. It is no accident that encyclopedic knowledge was included in some entries of the theoretically oriented Explanatory-combinatorial dictionary of Russian (Mel'chuk et al. 1984). On the other hand, many semantically-aware applications, including word sense disambiguation, semantic parsing, question-answering, textual entailment, etc. may also benefit from the availability of this information. Its potential is even stronger when we think about such knowledge-intensive tasks as commonsense reasoning, implicit knowledge extraction or bridging anaphora.

Of course, we are aware of the fact that creation of such resources for the language at large or even for its large fragment is extremely time- and effort-consuming. We would certainly prefer obtaining the information needed by some data-driven technique. However, ontological and semantic information extracted nowadays automatically out of large volumes of data is less than adequate for the tasks we are facing. We do not see any immediate prospect of automating this process and prefer to carry it out to the best of our abilities by the means we dispose of now. If future researchers find ways of automatically extracting such (or similar) information out of data, our resource may serve as the baseline.

We believe that onto-semantic descriptions of the type proposed in this paper are a useful step towards accumulating formalized knowledge. Our future efforts will be directed towards enlarging the stock of these descriptions and testing them in different applications.

Acknowledgements

The development of semantic resources described in this paper is a joint work being carried out by a team of lexicographers at the Institute for Information Transmission Problems, that consists of V. Dikonov, T. Frolova, E. Inshakova, L. Iomdin, A. Lazursky, I. Rygaev and S. Timoshenko. The format of descriptions presented here, as well as many concrete decisions were discussed at our weekly seminars. I am happy to thank all these colleagues for their active participation, constructive criticisms and excellent proposals.

This work was supported by the RSF grant 16-18-10422, which is gratefully acknowledged.

References

1. *Anisimovich K. V., Druzhkin K. Ju., Minlos F. R., Petrova M. A., Selegey V. P., Zuev K. A.* (2012) Syntactic and semantic parser based on ABBYY Compreno linguistic technologies. Computational Linguistics and Intellectual Technologies. Papers from the Annual International Conference "Dialogue", issue 11(18). Moscow. RGGU Publishers. p. 62–79
2. *Apresian Ju., Boguslavsky I., Iomdin L., Lazursky A., Sannikov V., Sizov V., Tsinman L.* (2003) ETAP-3 Linguistic Processor: a Full-Fledged NLP Implementation of the MTT. First International Conference on Meaning-Text Theory (MTT'2003). June 16–18, 2003. Paris: Ecole Normale Superieure, p. 279–288.
3. *Boguslavsky I., L. Iomdin, V. Sizov, S. Timoshenko.* (2010) Interfacing the Lexicon and the Ontology in a Semantic Analyzer. In: COLING 2010. Proceedings of the 6th Workshop on Ontologies and Lexical Resources (Ontolex 2010), Beijing, August 2010, pages 67–76.
4. *Boguslavsky I.* (2011) Semantic Analysis based on linguistic and ontological resources. Proceedings of the 5th International Conference on the Meaning—Text Theory. Barcelona, September 8–9, 2011. Igor Boguslavsky and Leo Wanner (Eds.), p. 25–36.
5. *Boguslavsky et al.* (2015)—Boguslavsky I., Dikonov V., Iomdin L., Lazursky A., Sizov V., Timoshenko S. Semantic Analysis and Question Answering: a System Under Development. In: Computational Linguistics and Intellectual Technologies. Papers from the Annual International Conference "Dialogue" (2015), p. 62–79.
6. *Boguslavsky I.* (2016) On the Non-canonical Valency Filling. GramLex 2016, co-located with COLING 2016. Proceedings of the Workshop on Grammar and Lexicon: Interactions and Interfaces, pages 51–60, Osaka, Japan, December 11.
7. *Cimiano Ph., Unger Ch., McRae J.* (2014). Ontology-based Interpretation of Natural Language. Synthesis Lectures on Human Language Technologies #24. Morgan & Claypool Publishers.
8. *Mairal Usón, R. y J. C. Periñán-Pascual.* (2009). The anatomy of the lexicon component within the framework of a conceptual knowledge base. Revista Española de Lingüística Aplicada 22, 217–244.
9. *Mel'chuk I., Zholkovsky A.* (1984) Explanatory-combinatorial Dictionary of Modern Russian. [Tolkovo-kombinatornyj slovar' sovremennogo russkogo jazyka]. Wien.
10. *Nirenburg, S., and V. Raskin.* (2004). Ontological Semantics. The MIT Press. Cambridge, Massachusetts. London, England.
11. *Pustejovsky J.* (1991). The Generative Lexicon. Computational Linguistics, vol. 17, N4, p. 409–441.