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

Moscow, May 31—June 3, 2017

GENERATION OF TEXT FROM ONTOLOGICAL SEMANTIC REPRESENTATION IN ETAP-3

Dikonov V. G. (dikonov@iitp.ru)

IITP RAS, Moscow, Russia

This paper describes an approach towards generation of text in natural languages from ontological semantic representations, produced by the linguistic processor ETAP-3. The system supports two distinct types of semantic representation, which use two different formal languages: Ontolanguage and UNL. They can be regarded conventionally as semantic sub-levels related to the framework of the Meaning↔Text theory. Here we present a comparison of both types of semantic representation and an overview of the process to convert ontological semantic structures into UNL graphs, which the system can turn into text in Russian and English.

Keywords: computational linguistics, functional model of language, linguistic ontology, semantic representation, text generation, RDF, UNL

ПОРОЖДЕНИЕ ТЕКСТА ИЗ ОНТОЛОГИЧЕСКИХ СЕМАНТИЧЕСКИХ ПРЕДСТАВЛЕНИЙ В ЭТАП-3

Диконов В. Г. (dikonov@iitp.ru)

ИППИ РАН, Москва, Россия

1. Introduction

ETAP-3 system [1] was build by the computational linguistics lab of IITP as a machine implementation of the Meaning↔Text theory by I. A. Melchuk (MTT) [8]. The theory defines multiple levels of representation for the information contained in any natural language (NL) text and suggests representation structures for every level, ranging from semantics to morphology. The levels are further split into "deep" and "shallow" sub-levels. Generation and analysis of text in this framework must follow a defined sequence of steps and create representation structures corresponding to all intermediate levels and their sub-levels. Until recently ETAP-3 implemented all necessary level-transition steps and provided a complete pipeline for both analysis and generation of text from morphology to deep syntax according to MTT and an intermediate syntactic-semantic representation in the Universal Networking Language (UNL) [9,10] format. This situation changed as a result of development of a new semantic analyzer called SemETAP [4]. The new module implements a different formal language for encoding semantic representations of natural language text based on an ontology. It is further called "Ontolanguage". The development of SemETAP involved introduction of a logical inference engine and inference rules into semantic analysis. The system can now produce semantic representations which are not completely derived from NL phrases, but contain additional statements expressed natively in a formal language. We also write formalized definitions of Russian word senses in the same fashion. Neither the results of logical inference nor the definitions are readable without specific training in computer science and linguistics and the system lacked capability to convert them to natural language.

Here we present an approach, which supports generation of text in different natural languages from any type of ontological semantic representation produced by the system. The process converts Ontolanguage representations into UNL and reuses existing modules of ETAP-3, which provide generation of text from UNL into Russian and English. UNL representation can also be processed by other UNL-aware systems to generate text in other natural languages. We consider Ontolanguage and UNL to be deeply related and treat them as successive stages in the text generation pipeline.

Sections 2–5 contain a brief introduction and comparison of both formal languages and corresponding semantic representations. Section 6 outlines the process of converting Ontolanguage semantic representation into UNL. Further processing is done by the UNL modules, which are beyond the scope of this paper.

2. UNL

UNL is designed to be an interlingua facilitating translation between different natural languages. It encodes the meaning of the source text in a way which is considered to be lossless. UNL code can be stored, processed and converted back into an equivalent natural language text without access to its source text. There are several dialects of UNL (UNL2005, UNL2010, U++, etc) [9, 10]. The dialect supported by ETAP3 is called U++ UNL.

The basic units of UNL are fine-grained concepts equal to word senses. They are derived from words of multiple natural languages. Definition of a UNL concept may incorporate stylistic/pragmatic nuances, which help to preserve the difference between senses of closely synonymous words and support exact translation. Such concepts are given unique labels called "Universal Words" (UW). The inventory of UNL concepts is defined by UNL dictionary, which can be easily enlarged to accommodate any number (even millions) of any new senses. UWs have links to words of many natural languages and more general ontology concepts, as explained in [6].

UNL represents text meaning as a directed graph, where nodes contain UWs. The graphs may include cycles (making them non-tree) and hypernodes (nodes containing other graphs instead of single UWs). The arcs are labeled with relations defined by UNL specs. There are 45 relations in UNL2005 and U++, however some changes were made among different versions of the specs. The set of relations includes:

- broadly defined semantic argument / deep case relations, such as agt "agent", ptn "partner", obj "patient, content of communication", ins "instrument", plc "place", plf "initial place", plt "target place", tim "time" etc.;
- two logical relations that replace the "and" and "or" conjunctions;
- relations of syntactic nature: mod "modifier" and man "manner, adverbial modifier", cnt for apposition, and a special relation to connect semantically independent parts of text.

The UNL dictionary [6] is also a knowledge base that provides semantic (equivalence, synonymy, antonymy, meronymy) and logical (subclass/instance) links between UNL concepts.

Each graph has a marked entry point, used to start interpretation during generation of NL sentences. The entry point usually corresponds with the syntactic head. A graph may contain multiple phrases, although our implementation always generates one graph per sentence.

UNL graphs are serialized (written as series of elementary statements) into sets of triplets, which follow the P(S,O) (predicate-subject-object) format. The nodes can carry UNL "attributes", used to encode syntactic categories, such as number, time and aspect, modality and attitudes of the speaker. UNL triplets can optionally be marked with a hypernode number, e.g. 01:P(S, O). Figure 1 shows an example of UNL graph and code.

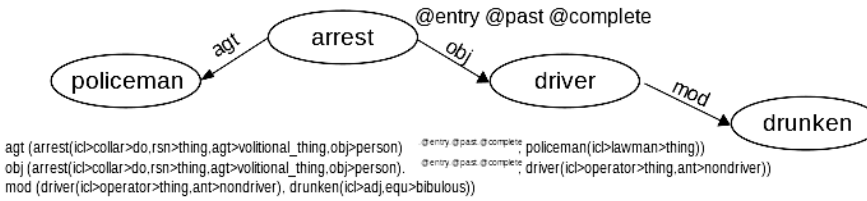


Fig. 1: UNL graph of "Полицейский арестовал пьяного водителя"/"Policeman arrested a drunk driver"

3. Ontolanguage

The ontology language serves a different purpose from that of an interlingua. It is geared towards common sense reasoning, fact extraction and question answering, but not translation. A semantic representation in the Ontolanguage is a graph stored in the standard RDF [11] format. Nodes of Ontolanguage graphs are concepts of the OntoETAP ontology [2], which is a fork of SUMO (Suggested Upper Merged Ontology) [7] in OWL with many changes and additions. The arc labels—predicates of RDF triplets—are OWL properties found in the same ontology. OntoETAP currently defines the complete inventory of concepts and arc types and expressive power of the language. Figure 2 shows the same phrase in the Ontolanhguage.

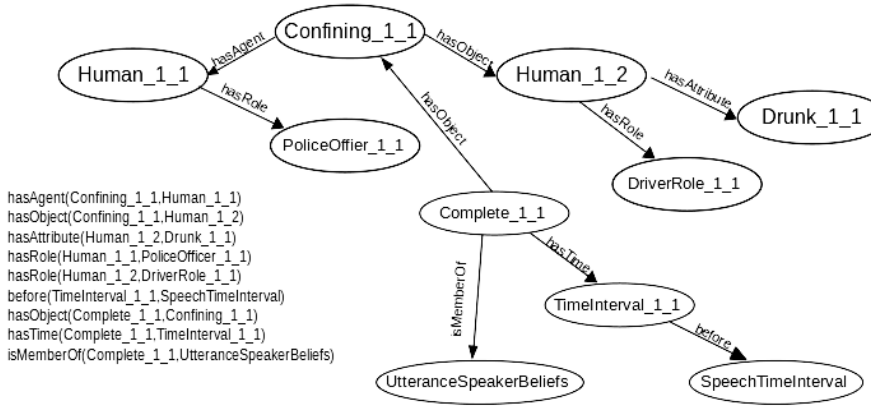


Fig. 2: Ontolanguage representation of "Полицейский арестовал пьяного водителя"/"Policeman arrested a drunk driver"

ETAP3 system supports building Ontolanguage graphs from Russian text only. The ontology semantic analysis module has its own memory bank (a DB), where it stores RDF statements. The DB is used to answer factual questions and support cross-sentence anaphora resolution. RDF graphs can be stored and processed by popular Semantic Web related tools, such as reasoners, SWRL rule and SPARQL query engines.

After processing a sentence the new RDF graph may be passed to an external tool which applies SWRL-like rules. The rules can fill-in some implicit information and replace complex concepts by subgraphs of basic concepts (semantic decomposition).

4. Comparison of the formal languages

A closer look at figures 1 and 2 reveals that the two semantic languages share a lot of important features both in technical and ideological planes.

4.1. Similarities

- Both formal languages encode meaning of natural language text.
- Both use the same basic formalism of directed graph (digraph). Both graph formats can contain cycles but self-cycles and parallel vertices are viewed as a defect. Little to none modification of the overall graph configuration is needed to translate from one formal language to the other.
- The graph nodes in both languages are filled with concept labels (ontology terms or UWs). Ontology concepts have exact equivalents in the UNL dictionary. New UWs can be added to translate any new ontology concepts, as long as they can be expressed in some human language.
- Both languages label graph arcs. Semantic argument / deep case labels with direct equivalents between the languages constitute the majority of arcs in almost any graph. Some examples are: agt—hasAgent, cag—hasAgent2, obj—hasObject, plc—hasLocation, etc.
- The graphs in both languages are serialized and stored in the form of PSO (predicate—subject—object) triplets.
- Both languages are extensible.

4.2. Differences

- Although the sets of arc labels in both formal languages overlap greatly, there are substantial differences. The set of arc labels in UNL is not extensible and it contains some syntactic relations as well as semantic ones (see section 2). The Ontolanguage has an open and extensible set of arc labels consisting of any OWL properties present in the OntoEtap Ontology.
- UNL format of triplets is more expressive than plain RDF, currently used by the Ontolanguage. 1) The S and O elements in UNL can have attribute tags P(S@attr1,O@attr2). They are used encode basic syntactic attributes, such as time, number, etc. (They are common for many NLs), modality and speaker attitudes. 2) A UNL triplet can have an optional 4th field containing a number. It indicates that the relation is a part of a numbered subgraph (hypernode) which functions as a single node.
- The attributes are just abbreviated forms of predicates and can be replaced by regular graph nodes, which makes them a convenience feature only. UNL hypernodes have no counterparts in plain RDF, but there are extensions, such as N-Quads [12], which make the expressiveness of both graph formats similar. One justification of using such extensions is the necessity to limit the scope of time/aspect/modality or negation statements.
- According to existing technical standards [11,13] ontology concepts in RDF are represented by Internationalized Resource Identifiers (IRI) which are global in the scope of the whole semantic web. An RDF graph can combine IRIs pointing to concepts of several ontologies and still be valid and completely interpretable. The current SemETAP module does not support this feature and relies on a single ontology only. Unknown IRIs that cannot be resolved to an ontology concept may cause errors.
- UNL concepts are limited by the scope of the dictionary of the corresponding UNL dialect. Unknown UWs may be parsed for links to known semantic classes.

5. Semantic structures

Although both formal languages are very similar from the technical viewpoint, corresponding semantic modules were designed to serve different purposes. As a result, they produce different semantic structures for the same source text.

UNL module of ETAP3 is designed to encode the meaning "as-is" without losses and intentional alterations of any kind. It never outputs anything beyond what is explicitly written in the source text. The SemETAP module, which uses the Ontolanguage, is geared towards creating a generalized but "complete" representation of the world described in the source text. It builds two kinds of structures: Basic Semantic Structure (BSemS) and Enhanced Semantic Structure (ESemS) [3]. In ESemS it attempts to fill in implicit information, which is supposed to be the common background knowledge of the speaker and the listener. Such expanded content may include common sense knowledge about the world and remembered contents of previously analyzed text.

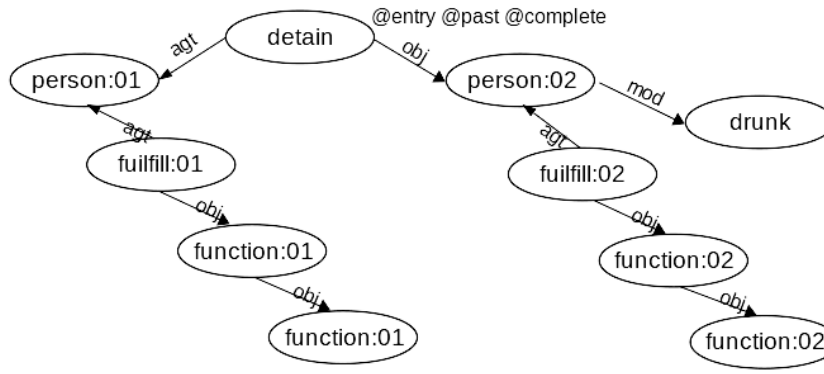
The two languages use different size of "semantic grain", which sets the maximal level of detail of the resulting semantic representations. It is a very important parameter, because a formal language for computer use is completely defined by its specifications (grammar) and associated existing resources (dictionaries, ontologies, tagsets, etc.). At the time of writing the OntoETAP ontology contained 7559 classes and 3742 named instances. The general domain part of the UNL dictionary [6] contained 92147 UWs, which is 8 times more. Most of the ontology concepts represent large classes of things/events and there are many UWs that represent subclasses and variations of such ontology concepts. E.g. *"Running"* is a terminal class in the ontology, but its UNL equivalent includes subclasses of *Jogging, Runnig at a moderate speed, Running at fast speed, Running Away, Fleeing,* etc. Different UWs linked with this concept also carry stylistic distinctions such as neutral *"run away / убегать"* vs colloquial *"scarper / сматываться"*.

The SemETAP module makes wide use of semantic decomposition. It replaces complex word senses by graphs of more general concepts. For example, the BSemS of the sentence in figure 2 uses three linked concepts to represent the sense of each of the words "policeman" and "driver": *hasRole(Human_1_1,PoliceOfficer)* and *hasRole(Human_1_2,DriverRole)*. The third concept is the predicate *hasRole*. Concepts *PoliceOfficer* and *DriverRole* are abstract attributes and cannot be used as the agent and object of *Confining.* Figure 3 shows a straight rendering of the result in English and Russian. Decomposition can compensate coarser semantic grain, but it is highly sensitive to completeness of underlying sense descriptions.

The UNL semantic structure has zero decomposition level. The difference between UNL and BSemS could be demonstrated by phrases like *Cop arrested speeder* and *Policeman arrested speeding driver*. UNL module captures the difference between *cop* and *policeman*, *speeder* and *speeding driver*, but SemETAP would ignore it as irrelevant and make identical structures. It shows that BSemS structure is in fact a product of normalization.

6. Generation of text

A semantic representation in the Ontolanguage is a generalization of the original Russian text with additional inferred statements. There are two possible approaches towards converting it to UNL. The first is to translate the source Ontolanguage structure and concepts in a most literal way. However, the text generated from UNL graphs made in this way will sound artificial. Figure 3 shows the result of converting the example from figure 2. This approach is useful for quick assessment of the source semantic representation contents and quality.



agt(detain(icl>do,equ>confine,agt>volitional_thing,obj>thing) @entry @past @complete; person(icl>organism>living_thing):01)
obj(detain(icl>do,equ>confine,agt>volitional_thing,obj>thing). @entry @past @complete; person(icl>organism>living_thing):02)
mod(person(icl>organism>living_thing):02, drunk(icl>adj,equ>intoxicated,ant>sober))
agt(fulfil(icl>carry_out>do,agt>person,obj>thing):01, person(icl>organism>living_thing):01)
obj(fulfil(icl>carry_out>do,agt>person,obj>thing):01, function(icl>duty>thing):01)
obj(function(icl>duty>thing):01, policeman(icl>lawman>thing))
agt(fulfil(icl>carry_out>do,agt>person,obj>thing):02, person(icl>organism>living_thing):02)
obj(fulfil(icl>carry_out>do,agt>person,obj>thing):02, function(icl>duty>thing):02)
obj(function(icl>duty>thing):02, driver(icl>operator>thing,ant>nondriver))

The person who fulfils the policeman's function had detained a drunk person who fulfils the driver's function.

Человек, который выполняет функцию полицейского, задержал пьяного человека, который выполняет функцию водителя.

Fig. 3: Results of converting the Ontolanguage graph of "Полицейский арестовал пьяного водителя" to UNL and further to English and Russian

Note that the concept *Drunk* is not expanded. Otherwise we might get a clumsy phrase like *The person who fulfills the policeman's function had detained a person who drinks alcohol and has slow reaction and behaves so that it causes danger to the person and other persons and fulfills the driver's function.*

An alternative approach is to imitate the human way of communication by deleting the expanded content of the source semantic structure and reverting certain known subgraphs back to complex concepts. This would make the resulting text much more natural: *Policeman detained a drunk driver.* Regardless of the approach chosen, the generated text (in any language) will not be an exact translation or equivalent of the source Russian sentence.

6.1. Conversion to UNL

The process of conversion from Ontolanguage into UNL is split into following steps:

- Optional compactification of certain subgraphs into single complex concepts by reversing the semantic decomposition rules[1], e.g. *hasRole(Human_1_1,Police Officer_1_1)* should become a single concept of *PoliceOfficer*, two reciprocal *Giving* events should become a single *Transaction* (*Exchange*) event, etc.
- Translation of conventional Ontolanguage statements corresponding to syntactic attributes of number, time, aspect, etc. and modality.
  - Ontolanguage introduces a special concept of a Set whenever UNL and NL languages simply use a plural marker.
  - Ontolanguage expresses time through references to the moment of speech or a universal timeline of the text. For example *hasObject(Complete_1_1,* Confining_1_1*), hasTime(Complete_1_1,TimeInterval_1_1), before(*Time Interval_1_1,*SpeechTimeInterval)* in Fig.2 show that the action is complete and took place in the past, which translates into UNL attributes @past and @complete.
  - Predicates of modality in the Ontolanguage must be converted into corresponding UNL modal attributes.
- Conversion of predicates in RDF triplets
  - Replace argument / deep case predicates with equivalents in UNL relations;
  - Complex predicates have to be expanded into small subgraphs, usually consisting of a single additional node. e.g. inUnit, hasPrice, hasRole.
- Translation of concept labels from ontology terms to UNL UWs.

This step concludes conversion of an Ontolanguage graph into a UNL graph. Further processing is done by UNL text generation modules.

Both formal languages are free from lexical ambiguity, so replacement of concept labels according to the dictionary works well. However, the problem of translation equivalent choice is not completely gone. According to UNL specifications (UNL2005, U++) [9, 10] UWs are split into formal classes roughly parallel to NL POS categories and closely related with argument frames:

- “thing” for objects and abstract notions is parallel with the noun category;
- “do” for actions of some agent, “be” for states of objects and “occur” for processes without an agent cover the verb category;
- “adj” for modifiers is similar to adjective category;
- “how” describes everything else.

Unlike NL POS types these categories do not impose specific requirements on the use of UWs in UNL, but they are respected in UW—NL word bindings and, therefore, define an important aspect of the future generated text syntactic structure. The choice between, e.g. writing(icl>activity>thing), write(icl>do,equ>spell,agt>perso

[1] This step is not implemented yet.

n,obj>information) will affect the resulting text, e.g "I think that writing/to write is tedious". The default for the ontology class Event (Process) is to use UWs of the do/be/occur categories, and UNL adj/how (modifier) categories for Attribute.

Implementation of this conversion process is in the proof-of-concept stage. The system can build a semantic structure from a Russian sentence and immediately transform it into a UNL graph that can be fed to any compatible UNL deconverter, including ETAP-3 itself. Parsing of externally created Ontolanguage graphs, including ones expanded by external inference tools, was not possible at the time of writing.

7. Conclusion

It has been demonstrated that two formal languages for semantic representations in ETAP-3 have a lot in common but the structures they encode are different. Basic Semantic Structures have higher degree of normalization than UNL structures, as shown in section 6. If we attempt to relate both types of structures with the MTT system of coordinates, UNL might be placed near the "shallow" end of the semantic level, while BSemS would be a candidate to a provisional "deep" or proper semantic sub-level. None of them is an exact implementation of the theory.

SemETAP module alone cannot be regarded as a true implementation of MTT in text analysis, because it does not handle the complete meaning of the source text. Direct transition from polysemic Russian words[2] to generalized concepts and lossy decomposition blur the meaning and cause permanent loss of some semantic information. The coarse grain of the BSemS structures is a design choice deeply rooted in the underlying linguistic and semantic resources. For example, the concept *Confining* completely blurs all difference between detaining, arresting, caging, locking-up (like in a wardrobe) and imprisonment. Lossless semantic decomposition is possible but writing the necessary definitions is very labor intensive. On the other hand, straightforward expansion of an ontology to accommodate large number of concepts bears a high computational complexity price for inference engines.

Conversion between the two semantic languages requires an extended concept hierarchy/ontology to link fine-grained concepts with general ones. Such approach is already implemented in UNL [6] and is advocated in [5] as Common Concept Hierarchy. Links between UNL and SUMO/OntoETAP allow easy transition from Ontolanguage to UNL. It fills a gap in the existing ETAP-3 text generation pipeline and allows to generate Russian and English text from arbitrary semantic representations (BSemS or ESemS).

The text produced by the text → ontological semantic representation → text[2] pipeline is not a translation. It reflects all semantic generalizations and additions introduced by the system. The resulting text shows the extent and precision of the "understanding" that the system can provide. This module, when completed, can be used in complex dialog systems, where a machine can generate a wide variety of messages and questions for a human user, approaching the goal of a sensible conversation with a computer.

---

[2] Internally ETAP-3 uses combinatorial dictionary, where entries called lexemes represent in fact sets of several related word senses. The system uses syntactic criteria for disambiguation which are not sufficient to distinguish between all senses.

Acknowledgements

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

References

1. *Apresjan Ju. D., Boguslavsky I. M., Iomdin L. L., Lazursky A. V., Sannikov V. Z., Sizov V. G., Tsinman L. L.* (2003) ETAP-3 Linguistic Processor: a Full Fledged NLP Implementation of the MTT; First Intemational Conference on Meaning-Text Theory (MTT'2003). Paris: Ecole Normale Superieur. P. 279–288.
2. *Boguslavsky I. M., Dikonov V. G., Timoshenko S. P.* (2012) An ontology for the support of tasks of extracting meaning fiom natural language text [Ontologiya dlya podderzhki zadach izvlecheniya smysla iz teksta na estestvennom yazyke]; Information technogies and systems. Moscow.
3. *Boguslavsky I. M., Dikonov V. G., Iomdin L. L., Timoshenko S. P.* (2013) Semantic representation for NL understanding; Proceedings of the International Conference "Dialogue". Issue 12(19), Moscow, RGGU Publishers. P. 132–144.
4. *Boguslavsky I. M., Dikonov V. G., Frolova T. I., Iomdin L. L., Lazursky A. V., Rygaev I. P., Timoshenko S. P.* (2016) Plausible Expectations-Based Inference for Semantic Analysis // Proceedings of the 2016 International Conference on Artificial Intelligence (ICAI"2016). USA: CSREA Press, 2016. P. 477–483. ISBN: 1-60132-438-3.
5. *Bhatt B., Poddar L., Bhattacharyya P.* (2013) IndoNet: A Multilingual Lexical Knowledge Network for Indian Languages; ACL.
6. *Dikonov V. G.* (2013) Development of lexical basis for the Universal Dictionary of UNL Concepts; Proceedings of the International Conference "Dialogue". Issue 12(19), Moscow, RGGU Publishers. P 212–221.
7. *Pease, A.* (2011) Ontology: A Practical Guide. Articulate Software Press, Angwin, CA. ISBN 978-1-889455-10-5.
8. *Mel'chuk L. A.* (1974) The Theory of Lingiustic Models of the type "Meaning↔Text" [Opyt teorii lingvisticheskikh modeley Smysl-Tekst]; Moscow, 1974 (2nd ed-1999).
9. UNL Specifications (2005), available at: http://www.undl.org/unlsys/unl/unl2005/.
10. U++ UNL Specifications (manuscript).
11. RDF Specifcations, available at https://www.w3.org/standards/techs/rdf#w3c_all.
12. RDF N-Quads, available at https://www.w3.org/TR/n-quads/.
13. RDF https://www.w3.org/TR/rdf11-concepts/#section-IRIs.