CANONIZATION OF NATURAL LANGUAGE BY MEANS OF UNIVERSAL SEMANTIC CODE

Igor Boyko PhD

Belarus State University
Minsk, Belarus
E-mail: igor_m_boyko@hotmail.com

The article is about canonization of a natural language for representing knowledge in the reasoning computer systems. Reasoning is a very difficult problem and the most important component of artificial intelligence systems. Universal Semantic Coode (USC) is used as a tool of canonization for knowledge representation and reasoning. There is no alternative yet in this kind of development and it is demonstrated in the article.

Key word: natural language processing, universal semantic code, semantics, knowledge representation, reasoning, canonization, action classification.

INTRODUCTION

Before we start talking about language canonization with USC, we have to analyze a notion of canonization toward natural language. Generally speaking, canonization or canonicalization is a process for converting data that has more than one possible representation into a "standard", "normal", or “canonical form”. (Wikipedia, https://en.wikipedia.org/wiki/Canonicalization).

Standard morphological canonization is lemmatization. (Wikipedia, https://en.wikipedia.org/wiki/Lemmatisation). In computational linguistics, lemmatization is the algorithmic process of determining the lemma for a given word where lemma is the canonical form, dictionary form, or citation form of a set of words. (Wikipedia, https://en.wikipedia.org/wiki/Lemma_(morphology)). For example, the words: “moves”, “moved”, “moving” after lemmatization are transformed into canonical form “move”.

Next kind of canonization is morpho-syntactic variations, which (largely) represent the same predicate and are semantically equivalent. For

example, “X composes Y” can be expressed also by “Y is composed by X” or “X does composition of Y” (Dagan, 2008). The phrase with infinitive can be considered as a canonical form of the phrase variations.

Finally, the most important and difficult kind is semantic canonization which includes ellipsis removal and missing phrase components restoration. In linguistics, ellipsis or elliptical construction refers to the omission from a clause of one or more words that are nevertheless understood in the context of the remaining elements. (Wikipedia, https://en.wikipedia.org/wiki/Ellipsis_(linguistics)). Ellipsisity is determined by language economy, however, the process is spontaneous and unordered (Martynov, 2009). Till this time, there are numerous distinct types of ellipsis acknowledged in theoretical syntax but there is no completed and consistent list.

In the article we discuss that semantic canonization is not limited with classic understanding of ellipsis and includes other important components. Semantic canonization today is only way to build intellectual systems which we could call as Artificial Intelligence (AI) systems. Without language meaning understanding and without reasoning on the basis of meaning understanding no one computer system can pretend to be AI system.

To represent knowledge, implement semantic canonization and provide knowledge reasoning we use USC (Martynov, 2001).

USC ACTION CLASSIFIER

1. Actions of first level

A definition: the **action of first level** is the action which operates only with physical or informational object.

USC semantic classifier is a universal semantic tool for semantic canonization. The classifier provides classification of actions represented by verbs and paired with their formal representation and canonical interpretation of each formal USC string and followed with definition of the action. For example, the action “insert” is represented by string ((XY)Z)((ZY)W) and its canonical interpretation is “X by means of Y inserts Z into W” and definition “to put or introduce into something”. There are 108 USC action classes. The action “insert” is a name of the class and has a corresponding set of action-analogous as members of the class. For example, the actions “embed”, “infix”,

“introduce” are actions-analogous and members of the class. So the USC string and its canonical reading stay the same for all of the action-analogous because they are pointed to the more abstract action and the name of the class “insert”.

This is not pure synonymic relations of actions in the USC classifier but mostly ontological relations. For example, the class “change” has in the set of action-analogous the actions “increase” and “decrease” which are, obviously, not synonyms.

Besides, in the USC string defined roles of the members of the action: “**subject** X by means of **instrument** Y inserts **object** Z into **mediator** W”. There are no more than four possible roles in the UCS string.

How USC canonization works we start showing on the example of the phrase: “The child eats with his hands”. Actually, to understand this phrase we use hidden inside of every human a model of the world. The model of the world (hidden knowledge) is architecture of patterns, i.e. the ordered set of patterns and the ordered set of transformations of some patterns in others. (Gordey, 2014).

The action “eat” is a member of the class “insert”. After substitution the roles from the initial phrase we can compile a canonized phrase: “The child by means of hands inserts food into the mouse”. We were forced to complete the phrase with missing data that we could guess on the ground of the model of the world. Now, there is a reasonable question: Does this phrase correspond to the meaning of the action “eat”? Probably not, more correctly would be to compile: “The child by means of mouse inserts food into the stomach.”

So, finally, what does it mean to eat with hands? Common sense tells us it means to eat holding food in hands. Only our model of the world allows understanding that. But the next raising question is: How a computer is going to understand all the logic to really satisfy to requirements for AI systems? Besides, it seems here the problem is not limited with only ellipsis revealing but includes the work around Subject-Action-Object (SAO) on the level of formal representation to define the structure of the SAO and implement some reasoning. Our answer is: USC is the tool providing some solution toward building AI systems.

Concerning SAO there is a gap between USC and pure linguistic approaches. For example, according to the approach of syntactic alternations (Levin, 1993) in the phrase “Nick broke the window” the ‘subject’ is Nick, but in the phrase “The window was broken” the ‘subject’ is the window. Such situation by default may not occur using USC because the action “brake” has

the formal representation defining obligatory roles of the members of the action where the window always will be in the position of the object. This knowledge corresponds to common sense and does not depend on syntactic structure of the sentence.

Full semantic reconstruction of some phrase is aimed to make all hidden or implicit information explicit. Explication of the sense is a way for evolution of AI systems (Martynov, 2009).

Continuing analysis of the phrase "The child eats food holding it in his hand" we have to reconstruct all SAO relations using USC classifier. That means deployment two actions "eat" and "hold" and filling them with appropriate roles.

So, the action "hold" is a name of the class represented by the USC string ((XY)Z)(Z(ZY')) having canonical interpretation "X by means of Y holds Z" and definition "to keep in a certain state, position". The action "insert" already has been considered. We can build full event explicitly: "The child by means of hands holds food. The child by means of hands inserts food into the mouse."

Of course, for the human such explication may seem redundant but without that effective knowledge representation to be loaded into the computer will not be possible.

2. Actions of second level

The **action of second level** is the action which operates only with another action.

The USC action classifier includes a list of actions not represented formally with USC string because they are not operating with physical or informational object. For example, the actions "activate", "intensify", "provide", "like" and many more are always in control of another following action and even if after such an action in the phrase you see an object it means that the following action was omitted and should be restored to determine an explicit fulfillment of the phrase.

For instance, the phrase "Activate your credit card now" means "Activate accessing to your credit card now." The phrase "I like this TV-show" means "I like watching this TV-show." As soon in the phrase the second order action is detected the alert for a missing following action, if it is missing, should be provided to build complete and explicit phrase. That constraint is necessary to

be implemented for better ability of a computer to communicate and to reason especially for intellectual problem solving.

For a human, such phrases sound very natural because of the model of the world. Even the best and powerful computer does not have any model of the world and will not have it until we build and load it in the memory of the computer.

A final example of language canonization is for the phrase: “A granddaughter writes to a grandfather.” Definitely “grandfather” is not ‘object’ here and it becomes clear after analysis of the action “write”. The action is a member of the class “encode” represented by the USC string ((XY)X)((XX)Y) having an interpretation “X bmo Y encodes W“ and a definition “to convert information into code“.

When we substitute roles we are forced again to introduce missing data using the model of the world and common sense knowledge: “The granddaughter by means of a pen encodes a letter”. Why does she do that? We can guess: to send it to the grandfather. So the complete event should be described as: “The granddaughter by means of a pen encodes a letter” and “The granddaughter intends to send the letter to the grandfather”. The action “intend” is a second level action and the actions “write” and “send” are the actions of the first level. The action “send” is a member of the class “approach” represented by the USC string ((XY)Z)((ZW)Z”) having the interpretation “X by means of Y approaches Z to W” and the definition “move toward something”.

Fully canonized phrase sound unnatural for a human but very explicit for a computer processing: “The granddaughter by means of a pen encodes a letter” and “The granddaughter by means of a mail approaches the letter to the grandfather”. The action “intends” can be dropped from consideration without losing the final meaning.

CONCLUSION

There are already dozens famous programs having extremely strict model of the world for playing: chess, jeopardy, go and others. But none of them are able to canonize natural language, solve problems of language semantics, and be called AI systems.

In conclusion, we would like to give a definition to semantic canonization of natural language. **Semantic canonization of natural language** is a process of reconstruction of the phrase or phrases including:

- restoration missing actions,
- restoration surrounding the action subject and object relations,
- determining the level of the action to be considered or dropped.

LITERATURE

Dagan I. Natural language as the basis for meaning representation and inference Computational linguistics and intelligent text processing / I. Dagan, R. Bar-Haim, I. Szpektor, I. Greental, E. Shnarch // A. Gelbukh (Ed.), 9th International Conference, CICLing 2008 : Haifa Israel. Springer. February 2008 Proceeding. P.151-170.

Kandelinsky S. Universal semantic code as an abstracting technology for fuzzy engineering problem solving. / S. Kandelinsky, I . Boyko. OSTIS-2014 (Open Semantic Technologies for Intelligent Systems) conference proceeding. Minsk. 2014. (in Russian)

Boyko I. Semantic classification of actions for knowledge inference. / I . Boyko. OSTIS-2016 (Open Semantic Technologies for Intelligent Systems) conference proceeding. Minsk. 2016. http://www.unsemcode.com

Gordey A. Theory of automatic generation of knowledge architecture. / A. Gordey. OSTIS-2014 (Open Semantic Technologies for Intelligent Systems) conference proceeding. Minsk. 2014. (in Russian)

Levin B. English verb classes and alternations. A preliminary investigation. / B. Levin. The university of Chicago press. 366 pages.

Martynov V. Foundations of semantic coding. Summary. / V. Martynov. European Humanity University. Minsk. 2001. http://www.unsemcode.com.

Martynov V. In the middle of conscious of the human. / V. Martynov. Belarus State University. Minsk. 2009. 272 pages. (in Russian)