Natural Language Text Generation from Knowledge Bases of ostis-systems

Artem Goylo
Intelligent Semantic Systems
Minsk, Belarus
Email: artemgoylo@gmail.com

Sergei Nikiforov
Belarusian State University of Informatics and Radioelectronics
Minsk, Belarus
Email: nikiforov.sergei.al@gmail.com

Olga Golovko
Brest State Technical University
Brest, Belarus
Email: olechkavgolovko@gmail.com

***Abstract*—The article discusses an approach towards generating coherent natural language texts from fragments of ostis-system knowledge bases. The architecture of the abstract sc-agent of translating a fragment of the knowledge base into a natural language is described. The task of generating a natural language text is subdivided into three sub-tasks: structure filtering, knowledge base fragment decomposition, and generating an equivalent natural language text. Two ways of linearizing a knowledge base fragment are proposed: one based on a predefined order specification and an algorithmic one. Generation of an equivalent natural language text is proposed to be done in two stages. Firstly, an intermediate rough natural language representation is generated using rule-based mappings between sc-text constructions and their corresponding natural language verbalizations. Secondly, the intermediate representation is converted into a coherent natural language text with the help of a large language model. Finally, three possible applications of the proposed approach are described.**

***Keywords*—Natural Language Generation, Natural Language Processing, semantic network, Open Semantic Technology for Intelligent Systems (OSTIS), SC-code (Semantic Computer Code), knowledge base, discourse structure**

I. Introduction

During the last couple of years we have seen a sharp improvement of AI-related technologies. With the rise of Large Language Models (LLMs), automatically generated natural language texts have reached a never-before-seen level of adequacy and coherence. However, as it has been well-established, LLMs are prone to hallucinations and factual distortions when generating responses [1]. One of the potential solutions to this problem is using a reliable source of information to provide verified information as context for an LLM (for example, see [2]). Knowledge bases can serve as such a reliable source of information.

The OSTIS Technology (Open Semantic Technology for Intelligent Systems) [3] is a technology of complex support of the next-generation intelligent computer systems development life cycle. The technology is particularly focused on using knowledge bases as the core of intelligent computer systems, which are called ostis-systems in the context of the technology.

The foundation of the OSTIS Technology is a universal means of semantic representation (coding) of information in the memory of intelligent computer systems, called sc-code. Texts in sc-code (sc-texts and sc-constructions) are unified semantic networks that have a basic set-theoretic interpretation. The elements of such semantic networks are called sc-elements (sc-nodes and sc-connectors, which, depending on whether they are directed or not, are called sc-arcs or sc-edges). The universality and unified nature of sc-code allow to use it in order to describe all kinds of knowledge and problem-solving methods, which, in turn, considerably reduces the difficulty of integrating methods and knowledge within a system as well as within a collective of such systems.

As sc-texts are, essentially, semantic networks, it follows that they are non-linear in their nature. This fact posits a particular challenge for the natural language generation task, since an sc-text needs to be linearized before being translated into a natural language text.

The goal of this work is to outline a module for generating natural language texts based on both complete and self-contained fragments of a knowledge base, as well as arbitrarily chosen fragments. This module is envisioned as part of the natural language interface of an ostis-system, described in greater detail in our earlier work [4]. To achieve this, we will have to address the following issues:

- Filtering irrelevant parts of an sc-text stored in the knowledge base;
- Linearization of a non-linear text that is a certain graph structure within the knowledge base;
- Translation of a linearized sc-text into a natural language.

II. State of the art

Automatic natural language text generation is a well-researched problem, with many different approaches having been proposed for solving it.

Traditional approaches usually based their natural language generation systems on the rules which were developed using the various theories of discourse structure, in particular, Rhetorical Structure Theory [5], which establishes 25 relations that bind sentences within a text segment. Such approaches usually formalize texts as trees whose nodes are specific text segments that are partitioned into smaller segments connected via a particular relation, with leaves of such trees being particular clauses. An example of an RST-based approach is [6]. Tree-based representation of texts is in line with such foundational discourse theories as [7].

The issue with the formalisms used by the traditional approaches is that both discourse macrostructure, as well as relations between individual sentences, are quite flexible and allow for a certain degree of individual variation. Besides, the subject domain of rhetorical relations at the micro-level is not completely formalized, and such relations can be numerous [6, p. 17], which complicates the development of natural language generation systems. On the other hand, some theories emphasize the impossibility of a complete account of relations that hold between text segments and propose a limited set of basic relations that are usually structural, rather than semantic or rhetorical [8].

Of the traditional approaches to natural language generation, our approach, which will be described below, is most similar to [9]. This approach utilizes a formal model of discourse strategies to generate coherent natural language texts. Prior to text generation, knowledge is extracted from a knowledge base, after which its relevancy is established and irrelevant parts are removed. The approach focuses on generating a coherent text as a response to a certain question, which is why it needs to establish relevancy of knowledge found in the knowledge base. The authors define three types of questions: questions about information available in the knowledge base, questions about definitions, and questions about differences between entities in the knowledge base [9, p. 7]. Text is linearized according to the chosen discourse strategy, and there is no predefined discourse structure available in the knowledge base [9, p. 8]. Such an approach allows to generate variable surface structures describing the same knowledge representation due to the focus on discourse strategies [9, p. 9]. Discourse structure is formalized as specific patterns of usage of rhetorical predicates (overall 16 such predicates are defined, e.g. attributive, equivalent, specification, explanation, evidence, analogy, etc.)

Modern approaches focus on using the latest achievements in the field of AI, in particular, neural networks. Among them, one popular way of generating natural language texts is by using some intermediate semantic representation: for example, formalisms of Discourse Representation Theory [10], [11]. An approach that is similar to ours is the variant of translating RDF-triples into a natural language text, proposed in [12].

However, a significant number of works within such approaches focuses on sentence-level generation, rather than document-level generation. At the same time, approaches that emphasize document-level generation [10], [11], though acknowledging the task of text linearization (i. e. ordering of text segments), focus more on solving particular problems at the sentence-level connected with the chosen semantic formalism, such as variable naming.

Still other neural network-based approaches do not use intermediate semantic representation in natural language generation, see, for example, [13]. However, pure neural approaches sometimes have to contend with the issue that high-level semantic relations, which are important to capture when generating a coherent natural language text, present a challenge for neural networks [13, p. 1]

III. Suggested approach

We propose to address the issue of generating a coherent natural language text by adopting a multi-agent approach to the design of the respective module of natural language generation based on the OSTIS Technology.

The foundation of a knowledge base developed using the technology is a hierarchical system of semantic models of subject domains and ontologies. An ostis-system problem solver is represented by a collective of agents (sc-agents) that interact with each other by specifying the information processes in the semantic memory that they execute [14].

An abstract sc-agent is a certain class of functionally equivalent sc-agents, different instances (i. e. representatives) of which can be implemented in different ways. [14]

Below we will describe our suggested approach by providing the specifications of agents required for solving the problem of translating a fragment of the knowledge base into a natural language text.

The proposed implementation of the agent of translating a fragment of the knowledge base into a natural language has the following decomposition:

Abstract sc-agent of translating a fragment of the knowledge base into a natural language
⇒ *abstract sc-agent decomposition**:
- {• *Abstract sc-agent of structure filtering*
- • *Abstract sc-agent of fragment decomposition*
- • *Abstract sc-agent of generating an equivalent natural language text*
  - ⇒ *abstract sc-agent decomposition**:
    - {• *Abstract sc-agent of generating a rough version of a natural language text*
    - • *Abstract sc-agent of converting the rough version into a correct natural language text*

}
}

The agents that are part of this abstract sc-agent will be discussed below. We will also discuss the agent of structure filtering that may be used in order to remove specified parts of a fragment before translating it into a natural language. The introduction of such an agent is explained by the fact that the structures that are to be translated (e. g., the semantic neighborhood of a certain concept) could be quite large and/or include irrelevant information, which would lead to a bloated natural language text that would be more difficult to comprehend.

A. Abstract sc-agent of structure filtering

The input of this agent is the structure to be filtered, and the output of it is a certain subset of the structure.

The filtering is done by using templates. Two kinds of template sets are proposed: the set of exclusive templates, i.e. the templates that are used to exclude a part of the structure before translation, and the set of inclusive templates, i. e. the ones that specify what part of the structure needs to be kept.

If a set of inclusive templates is passed to the filtering agent then only the corresponding part of the structure will be outputted. If exclusive templates are used then only the part that does not fit the template will be translated.

It is also necessary to introduce additional logic — the agent needs to check for connections between the elements of the part to be excluded and the part that is to be kept.

For example, let us consider the situation in figure 1.

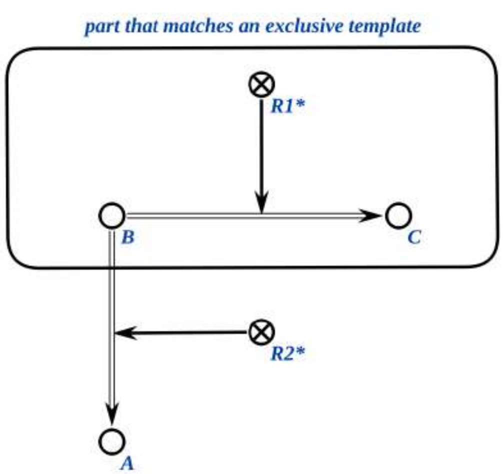

Figure 1. An example of a structure prior to filtering.

Here, it is expected that what will be left after filtering is the fragment of the structure in figure 2: even though the node *B* was part of the pattern found using an exclusive template, it still has a connection with the part that is to be included after filtering.

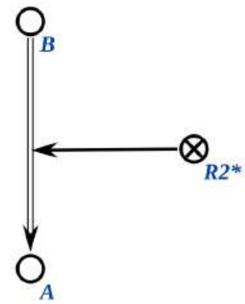

Figure 2. An example of the result of filtering.

B. Abstract sc-agent of fragment decomposition

The goal of this agent is decomposing a fragment of the knowledge base into an ordered set of substructures. The order of the substructures indicates the ordering of the content of each structure during after translation.

Classic works on discourse structure note that the structure of specific discourses depends on their genre (e. g., a story and a scientific paper will have different structures) [7]. Thus, different fragments of the knowledge base can have their own respective canonical structures. Hence it is important to discuss, which fragments of ostis-system knowledge bases are most likely to be translated into a natural language, and to formalize their structure.

The structure of a text depends on the class of the structure to be translated and the purpose of its translation, i. e. to what kind of message it can be used as a response.

The input of the agent is a structure to be translated, while the output is an ordered set of substructures that is the decomposition of the original structure.

This set of substructures is obtained in two stages:

- firstly, a predefined text structure specification is used;
- secondly, the ordering of elements within the substructures is derived algorithmically.

At the first stage, we propose to use a predefined specification of ordering of substructures stored in the knowledge base. The substructures are assumed to be the partition of the original structure (fragment) in the knowledge base. Such specifications can be defined for sc-texts of subject domains and other frequently used fragments of the knowledge base.

For example, we propose that a subject domain has a specification that includes the next elements in the following order: classes of objects of research (first, maximal classes and then non-maximal ones), explored relations, didactic relations (e.g. explanations and annotations), key signs, the list of segments that comprise the subject domain. After those elements comes the text of the segments themselves. The order of the segments is specified by the author of the subject domain specification, which eliminates the need to determine their order automatically.

Having such a specification will improve the quality of the resulting natural language text. However, in order to allow for translation of an arbitrarily defined fragment of the knowledge base, as well as to define the order of elements within the aforementioned substructures, an algorithmic way of deriving the order is needed.

However, the elements listed above (substructures of the original fragment of the knowledge base (in this case, the sc-text of a subject domain)) contain elements of their own, which also need to be linearized. Therefore, at the second stage, we derive the order of elements within such substructures based on the concepts contained in them. This process includes:

- Derivation of the order of concepts in the structure to be translated, i. e., the order in which their semantic neighborhoods should be translated;
- Derivation of the order of elements within such semantic neighborhood.

At the first step we propose to build a tree (graph) of dependencies between concepts according to the relations between them, and then to use this tree to derive the order of elements. For example, if a fragment has multiple classes of objects, then the first to be translated should be the supersets, followed by the subsets; sets should be translated before their elements, and so on.

At the second step (derivation of the order of elements within the semantic neighborhood of each concept) we propose to use a predefined order of relations and parameters. This will allow us to specify, for example, that when translating the semantic neighborhood of a concept the first elements to be translated should be the concept's identifiers, then its definition, then its membership in different sets followed by all the subsets of the given concept. There can be multiple potential variants of such a specification, depending on the class of the fragment.

We should note that this agent, and the structure specifications used by it, can be utilized not only for translating fragments of the knowledge base into a natural language, but also for translating them into other variants of linear representation of sc-code, for example, SCn.

C. Abstract sc-agent of generating an equivalent natural language text

As mentioned above, this agent in turn includes the following agents:

- Abstract sc-agent of generating a rough version of a natural language text
- Abstract sc-agent of converting the rough version into a correct natural language text

The input of the first agent is a structure to be translated, while the output is an ostis-system file with the resulting text. The text obtained as a result of this agent's execution may not fully correspond to the grammar of a particular natural language (in our case, English).

This approach explicitly sidesteps a much more complex task of microplanning (i.e. mapping of certain information in the semantic representation to the verbalization of this information) [12]. Instead of solving the problem of generating referring expressions, lexicalization, and so on, at this stage we propose to use a straightforward approach of using a finite set of specific rules of translating sc-code expressions into a natural language.

Currently, the agent has a simplified variant of implementation that is reduced to implementing a number of translators, each of which is dedicated for processing certain sc-code constructions (e. g., parameters of elements, their relations, etc.) Every construction has a corresponding natural language verbalization that is used during the translation. An example of a construction to be translated can be seen in figure 3.

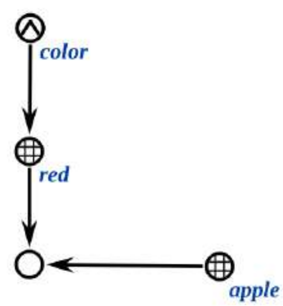

Figure 3. An example of a construction to be translated.

The membership arc corresponds to the English *is*, while the parameter in this case is translated as a pair (parameter, value). Therefore, the construction above will be translated by this agent into the following rough natural language text: *is apple, color red.*

This implementation has been chosen because it is relatively uncomplicated. In the future, we plan to elaborate it in way that the agent would arrive at a rough natural language verbalization algorithmically using formalization of natural language syntax proposed in our earlier work [15].

Complete rule-based algorithmic translation of knowledge base fragments into an adequate and coherent natural language text appears to be an infeasible task due the complexity of decision-making at each stage of the process. This is exemplified by the fact that the practice of designing fully rule-based intelligent systems has been largely supplanted by application of neural network-based solutions, which, in the case of large language models, outperform all currently available methods of automatic natural language text generation.

It is for this reason that we propose to introduce the *Abstract sc-agent of converting the rough version into a correct natural language text*. This agent is implemented using a large language model. Its input and output arguments are ostis-system files. The input file contains the original rough natural language text that needs to be transformed, and the output file contains the text generated by a large language model.

Using a large language model is a convenient alternative to rule-based generation because it allows to sidestep certain sub-tasks of generating a coherent natural language text, such as choosing particular means of cohesion and coherence, which allows us to reduce the task to forming an ordered set of substructures that sets the order of segments of a coherent text, while individual verbalization choices are made by the large language model, which they in general excel at [16].

Using an intermediate representation for now also increases the likelihood of obtaining acceptable results without language model hallucinations [1], since the model in this case is not utilized in a zero-shot scenario and is provided with an extensive context that has a formal nature.

Thus, to generate the resulting natural language text we propose to use and intermediate representation (the output of the agent of generating a rough version of a natural language text), which is necessary at this stage because existing neural network solutions cannot be directly integrated with knowledge bases of ostis-systems. In the future, the OSTIS Technology will have support for "native" representation of neural networks as well as the means of preprocessing the input for traditional neural networks in such a way as to enable them to handle sc-code constructions [17]. This will eliminate the need for translating fragments of the knowledge base into intermediate variants of representation, and will enable us to use the actual sc-text of a knowledge base fragment as input for a large language model.

IV. Potential applications

Finally, we would like to discuss potential applications of the natural language generation module described above. These are three main ways in which it can be used:

- Exporting an arbitrary fragment of the knowledge base in a natural language;
- Navigating the knowledge base in a natural language;
- Dialog with an ostis-system using a natural language.

A. Exporting an arbitrary fragment of the knowledge base in a natural language

In this scenario, the fragment to be exported is specified by the user manually. For this application, the corresponding ostis-system can support various existing natural language text formatting styles.

One potential benefit of translating arbitrary fragments of the knowledge base into a natural language is that it makes it possible to use knowledge bases appropriately as the primary means of storing knowledge. Whereas, traditionally, knowledge has been stored mostly in natural language texts of various kinds, having a system that allows to translate formalized representations of knowledge into natural language texts on demand will significantly help with complex automation of various types of human activity [18].

B. Navigating the knowledge base in a natural language

The main way to navigate he current OSTIS Metasystem interface [19] is by navigating semantic neighborhoods of elements and/or other constructions. The external languages of sc-code representation used for outputting the content of the Metasystem's knowledge base are SCn and SCg [3].

It is possible to introduce a new way of navigating knowledge bases of ostis-systems whereby the fragments are translated into a natural language, which makes it possible to interact with ostis-systems effectively for users who are unfamiliar with the languages of external representation of sc-code.

This application would require additional work on the translation module in order to allow for hyperlinks within the natural language text markup, which will make the navigation easier.

C. Dialog with an ostis-system using a natural language

We plan to provide for the ability to communicate with an ostis-system using a natural language by implementing a question-answering support subsystem for users of the OSTIS Metasystem [19]. This subsystem should allow the user to ask questions about any knowledge stored in the Metasystem's knowledge base and get a response in a natural language.

The pipeline of this subsystem can be decomposed into the following stages:

- Message classification and question argument identification;
- Response generation;
- Translation of the response into a natural language using the means described above.

During the ongoing implementation of the prototype of this subsystem we have decided to use one of the existing neural network-based classifiers for the task of message classification and question argument identification: Rasa [20], Wit.AI [21], and others. We consider

Rasa to be the preferable option due to the possibility of local deployment and its open-source nature.

This approach has been chosen in order to obtain quickly a working prototype of the system. In the future, neural network-based classifiers can be replaced with an sc-agent of natural language understanding based on the approach discussed in [15].

The input of the response generation agent is a message that has been classified, while the output is a structure from the knowledge base that is an appropriate response to the message. An example of message classification received by the agent is available in figure 4.

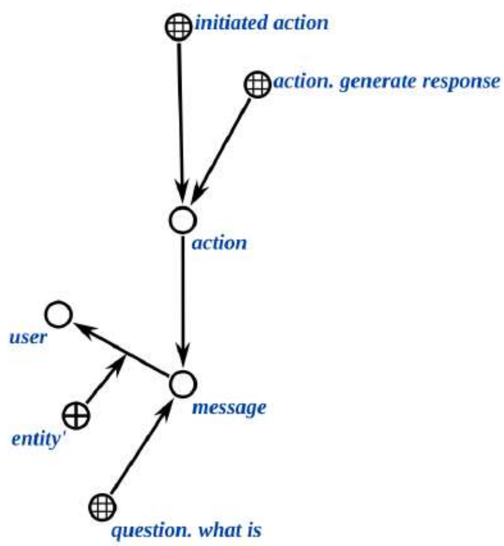

Figure 4. An example of message classification.

The agent operates in two steps:

- firstly, it tries to formulate a response using search templates;
- secondly, it tries to formulate a response by executing an appropriate action, in case the first step was unsuccessful.

The first step is introduced because responding to certain user questions can be reduced to searching for a relatively simple construction in the knowledge base, which can be implemented by mapping the corresponding classes of questions to certain search templates, as well as mapping message arguments to variables contained in such search templates. The response is a structure that contains the result of search by a template that corresponds to a certain class of questions after variables have been replaced with the corresponding question arguments.

Extending the set of questions that can be answered using search templates is an uncomplicated task that does not require modifying the problem solver. This task is reduced to introducing a new search template and specifying its connection with a class of questions and its arguments.

However, such constructions may be difficult to describe using one search template, or the answer may not be reducible to simple search and may require detailed transformations. For this reason, the second step is necessary.

If formulating a response using search templates is impossible, then the system searches for classes of actions connected to the corresponding class of questions by the relation *response action**. An instance of such action is then created with the corresponding argument received in the question.

Let us illustrate this using the question *What is X?* as an example. The response to such questions is a description of a certain element in the knowledge base, i. e. its semantic neighborhood. An example of the connection between a class of actions and a class of questions described above can be seen in figure 5.

In order to handle questions with two or more arguments that have different roles, the roles in the message can be mapped to respective arguments of actions in the knowledge base.

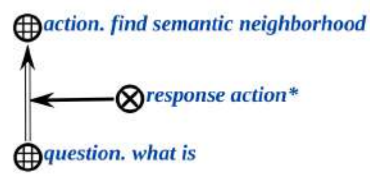

Figure 5. An example of the mapping between a class of actions and a class of questions.

Thus, when the agent receives the message illustrated in figure 4, an instance of the action of finding a semantic neighborhood is created. Then the problem solver waits until the agent is executed, and the agent's response is then connected with the message. An example of a construction obtained in this way can be seen in figure 6.

V. Conclusion

We have provided a sketch of the architecture of a module for translating fragments of ostis-system knowledge bases into coherent natural language texts. Our proposed approach subdivides the task of generating a natural language text into three sub-tasks: structure filtering, knowledge base fragment decomposition, and generating an equivalent natural language text.

The most important sub-task is knowledge base fragment decomposition because it ensures cohesion and coherence of the resulting natural language text. We have proposed two preliminary ways of solving this task: specification of element ordering within a fragment of the knowledge base as a sort of schema of the overall structure of the resulting natural language text, and

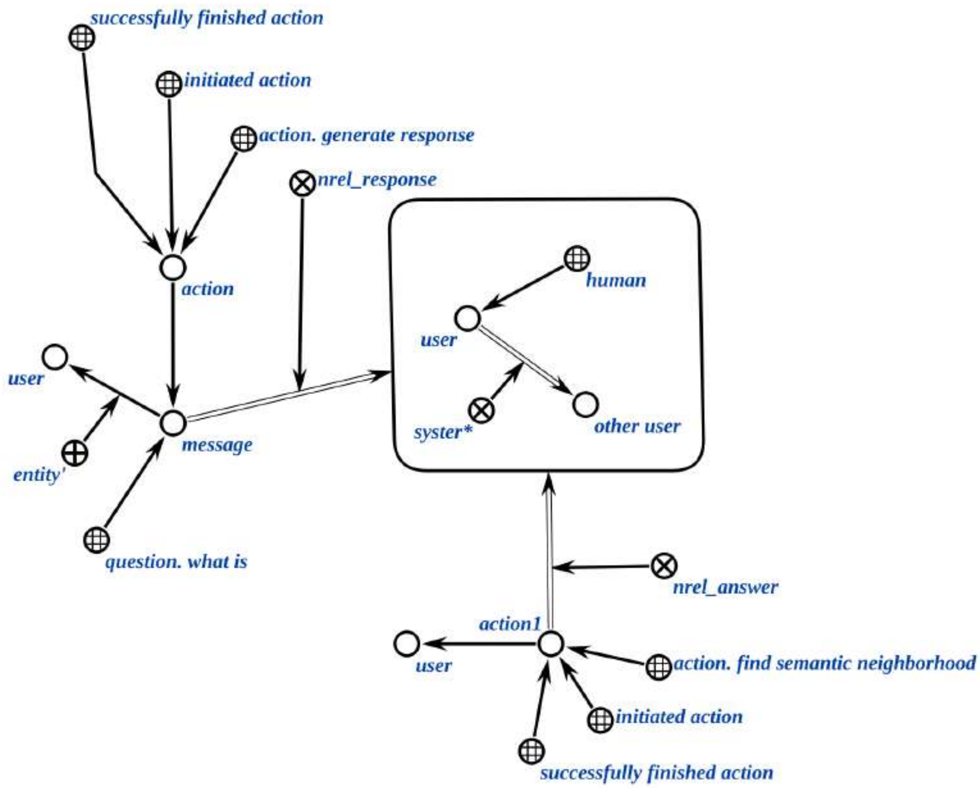

Figure 6. An example of the output of the response generation agent.

algorithmic approach that builds a tree of dependencies between certain relations.

The actual generation of a natural language text is proposed as a two-step process, whereby a large language model generates the resulting text from an intermediate representation.

Due to the preliminary character of our work, there are certain limitations. Our approach does not discuss linearization of graph-based formal texts in greater detail apart from positing relatively straightforward schemata to be used during translation. In fact, given that the ultimate application of the module discussed here is natural language dialog between humans and intelligent systems, our proposed approach can be improved in three different ways:

- Firstly, understanding natural language questions to the system can be done using not a simple classifier but rather a combination of syntactic and semantic analysis modules that use a formalization of natural language syntax and semantics.
- Secondly, the linearization task can be solved in a much more elaborate manner. This would require formalization of a discourse structure model within the knowledge base of an ostis-system. The model can then be used to intelligently derive macro- and microstructures of sc-texts to be translated into a natural language.
- Finally, the actual translation of linearized sc-texts into a natural language needs further elaboration. An obvious improvement is to eliminate specific rules (mappings) of translating sc-constructions into certain predefined natural language verbalizations, which would require designing a proper natural language synthesis module.

All of the above can be considered as potential directions for future research.

Acknowledgment

The authors would like to thank the research teams of the Department of Intellectual Information Technologies of Belarusian State University of Informatics and Radioelectronics and the Department of Linguistics Disciplines and Cross-Cultural Communication of Brest State Technical University for their help and valuable comments.

References

[1] T. Yu, S. Zhang, and Y. Feng, "Truth-aware context selection: Mitigating the hallucinations of large language models being misled by untruthful contexts," 2024. [Online]. Available: https://api.semanticscholar.org/CorpusID:268364338

[2] Y. Wu, N. Hu, S. Bi, G. Qi, J. Ren, A. Xie, and W. Song, "Retrieve-rewrite-answer: A kg-to-text enhanced llms framework for knowledge graph question answering," 2023.

[3] V. Golenkov, Ed., *Tehnologija kompleksnoj podderzhki zhiznennogo cikla semanticheski sovmestimyh intellektual'nyh komp'juternyh sistem novogo pokolenija [Technology of complex life cycle support of semantically compatible intelligent computer systems of new generation].* Bestprint, 2023.
[4] A. Goylo and S. Nikiforov, "Natural language interfaces of next-generation intelligent computer systems," *Open semantic technologies for intelligent systems*, no. 6, pp. 209–216, 2022.
[5] W. Mann and S. Thompson, "Rethorical structure theory: Toward a functional theory of text organization," *Text*, vol. 8, pp. 243–281, 01 1988.
[6] E. H. Hovy, "Automated discourse generation using discourse structure relations," *Artificial Intelligence*, vol. 63, no. 1, pp. 341–385, 1993. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0004370293900213
[7] W. Kintsch and T. A. van Dijk, "Toward a model of text comprehension and production." *Psychological Review*, vol. 85, pp. 363–394, 1978. [Online]. Available: https://api.semanticscholar.org/CorpusID:1825457
[8] B. J. Grosz and C. L. Sidner, "Attention, intentions, and the structure of discourse," *Computational Linguistics*, vol. 12, no. 3, pp. 175–204, 1986. [Online]. Available: https://aclanthology.org/J86-3001
[9] K. R. McKeown, "Discourse strategies for generating natural-language text," *Artificial Intelligence*, vol. 27, no. 1, pp. 1–41, 1985. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0004370285900827
[10] J. Liu, S. Cohen, and M. Lapata, "Text generation from discourse representation structures," 01 2021, pp. 397–415.
[11] C. Wang, R. van Noord, A. Bisazza, and J. Bos, "Evaluating text generation from discourse representation structures," in *Proceedings of the 1st Workshop on Natural Language Generation, Evaluation, and Metrics (GEM 2021)*, A. Bosselut, E. Durmus, V. P. Gangal, S. Gehrmann, Y. Jernite, L. Perez-Beltrachini, S. Shaikh, and W. Xu, Eds. Online: Association for Computational Linguistics, Aug. 2021, pp. 73–83. [Online]. Available: https://aclanthology.org/2021.gem-1.8
[12] C. Gardent, A. Shimorina, S. Narayan, and L. Perez-Beltrachini, "The WebNLG challenge: Generating text from RDF data," in *Proceedings of the 10th International Conference on Natural Language Generation*, J. M. Alonso, A. Bugarín, and E. Reiter, Eds. Santiago de Compostela, Spain: Association for Computational Linguistics, Sep. 2017, pp. 124–133. [Online]. Available: https://aclanthology.org/W17-3518
[13] J. Guan, X. Mao, C. Fan, Z. Liu, W. Ding, and M. Huang, "Long text generation by modeling sentence-level and discourse-level coherence," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, C. Zong, F. Xia, W. Li, and R. Navigli, Eds. Online: Association for Computational Linguistics, Aug. 2021, pp. 6379–6393. [Online]. Available: https://aclanthology.org/2021.acl-long.499
[14] D. Shunkevich, "Hybrid problem solvers of intelligent computer systems of a new generation," *Open semantic technologies for intelligent systems*, no. 6, pp. 119–144, 2022.
[15] A. Goylo and S. Nikiforov, "Means of formal description of syntax and denotational semantics of various languages in next-generation intelligent computer systems," *Open semantic technologies for intelligent systems*, no. 6, pp. 99–118, 2022.
[16] L. Shu, L. Luo, J. Hoskere, Y. Zhu, Y. Liu, S. Tong, J. Chen, and L. Meng, "Rewritelm: An instruction-tuned large language model for text rewriting," 2023.
[17] M. Kovalev, A. Kroshchanka, and V. Golovko, "Convergence and integration of artificial neural networks with knowledge bases in next-generation intelligent computer systems," *Open semantic technologies for intelligent systems*, no. 6, pp. 173–186, 2022.
[18] V. V. Golenkov, N. A. Gulyakina, and D. V. Shunkevich, "Klyuchevye problemy i strategicheskie tseli rabot v oblasti iskusstvennogo intellekta [key problems and strategic goals of research in the field of artificial intelligence]," 2023, pp. 9–13.
[19] "ostis-metasystem repository," Available at: https://github.com/ostis-ai/ostis-metasystem, (accessed 2024, March).
[20] "Rasa," Available at: https://rasa.com/, (accessed 2024, March).
[21] "Wit AI," Available at: https://wit.ai/, (accessed 2024, March).

ГЕНЕРАЦИЯ ЕСТЕСТВЕННО-ЯЗЫКОВЫХ ТЕКСТОВ ИЗ БАЗ ЗНАНИЙ OSTIS-СИСТЕМ

Гойло А. А. Никифоров С. А. Головко О. В.

В статье описывается подход к генерации связных текстов на естественном языке из фрагментов баз знаний ostis-систем. Описана архитектура абстрактного sc-агента трансляции фрагмента базы знаний в естественный язык. Задача генерации естественно-языкового текста подрязделяется на три подзадачи: фильтрация структуры, декомпозиция фрагмента базы знаний и генерация эквивалентного естественно-языкового текста. Предлагаются два способа линеаризации фрагментов базы знаний: использование заданной спецификации порядка элементов фрагмента и алгоритмический. Предлагается выполнять генерацию эквивалентного естественно-языкового текста в два этапа. На первом этапе формируется черновое естественно-языковое представление на основе правил сопоставления конструкций sc-текстов с соответствующими им естественно-языковыми формулировками. На втором этапе такое промежуточное представление транслируется в связный естественно-языковой текст с использованием большой языковой модели. Описываются три возможных применения предлагаемого подхода.

Received 18.03.2024