DOI: 10.17323/2587-814X.2024.1.7.21

Development of an intelligent assistant for selection of goods in the process of dialogue with the user

Dmitry E. Palchunov [a]
E-mail: palch@math.nsc.ru

Alexander A. Yakobson [b]
E-mail: a.yakobson@g.nsu.ru

[a] S.L. Sobolev Institute of MathematicS, Siberian Branch of the Russian Academy of Sciences
Address: 4, Akademika Koptyuga Ave., Novosibirsk 630090, Russia

[b] Novosibirsk National Research State University
Address: 1, Pirogova St., Novosibirsk 630090, Russia

Abstract

This article is devoted to the development of methods for creating intelligent assistants. Intelligent assistants can be used in call centers to solve customer problems, to solve technical support tasks, to help people with disabilities, to help in choosing goods, etc. We consider intelligent assistants that engage in argumentative dialogue with users, aimed at finding goods and services that maximally satisfy users' wants and needs. The development of the intelligent assistant is based on a four-level model of the subject domain and a semantic model of the user. The system under development automates the

* The article is published with the support of the HSE University Partnership Programme

process of search and decision justification through the reuse of domain cases: accumulated knowledge about previous dialogues with users. This gives the system we developed an advantage over existing analogues, which are incapable of reusing knowledge about previous dialogues. The paper develops a case-based approach to building an intelligent system capable of reasoning about its responses. For this purpose, an argumentation graph is constructed, methods for structuring domain cases are developed, and ontological homomorphisms are used to transform the available domain cases into a finished solution. A description of model-theoretical methods for constructing intelligent assistants is presented. The cases of goods, users and dialogues of an intelligent assistant with users are formally described in the form of partial models. The transformation of domain cases and similarity of cases are formalized using ontological homomorphisms of partial models. The purpose of the developed dialogue system is not only to select a solution according to the user's request, but also to find out the tasks that the user is going to solve, to analyze his argumentation, and then to justify the proposed solution to the user, to show that this particular product or service will be able to meet his needs.

Keywords: intelligent assistant, argumentative dialogue, domain case, partial model, ontological homomorphism, ontological model of the subject domain, semantic model of the user

Citation: Palchunov D.E., Yakobson A.A. (2024) Development of an intelligent assistant for selection of goods in the process of dialogue with the user. *Business Informatics*, vol. 18, no. 1, pp. 7–21.
DOI: 10.17323/2587-814X.2024.1.7.21

Introduction

The world is currently experiencing the peak of popularity of artificial intelligence (AI) technologies. The success of ChatGPT as a universal dialogue system has shown the need of people to have an intelligent tool for solving various tasks. ChatGPT solves many tasks: from writing texts for social networks (with great success) to creating scientific papers (with not such great success).

In most cases, such systems act as extremely advanced and powerful content compilers; they search through existing data and piece by piece assemble the required result from them. At the same time, the very concept of neural networks imposes a very specific limitation on them: they rely on their own trained model, on the data entered into the training sample in advance. As a result, they cannot use recent results of their own work to improve the process of finding a solution, because retraining a neural network is a long and resource-intensive process. Also, if any fundamentally new object appears in the subject domain (for example, a new style of drawing, if we are talking about a neural network that creates images), the neural network will not be able to obtain the same result on its own, because this new data was not incorporated into it during training.

The situation is additionally complicated by the fact that neural networks are a "black box." It is practically impossible to interpret the process of their work, especially for large industrial instances. One of the possible solutions to the problems of the neural network approach is the construction of a logical system based on semantic analysis and structuring domain cases: previous sessions of work of the intelligent system both with this user and with all previous users.

1. Tasks of the intelligent assistant

Our goal is to develop methods for reasoned dialogue between an intelligent assistant and a user in order to help the user achieve goals, realize his/her intentions, satisfy needs and solve problems. In this paper, we primarily consider intelligent assistants that help users in selecting appropriate products and services.

To achieve this goal, the following tasks need to be performed in an automated manner:

- identifying the user's interests, needs, desires, goals and intentions;
- finding out how the user achieves their goals, solves their tasks, fulfils their intentions and satisfies their needs (e.g. the intention to purchase a desired product);
- identifying the user's justification, explanation, reasoning why, for example, he/she needs this particular device; finding out the specific tasks that the user is going to perform with this device (e.g. viewing and editing photos, cleaning the room or controlling a smart home);
- selecting for the user the product or service that best suits the user's tasks and needs;
- building an argumentation, justifying that a given product or service is indeed the best for the user (subject to the fulfilment of product price constraints and other non-functional requirements), or offering the user a set of products that are best suited for solving their problems;
- explanation of the differences between these products, their positive and negative qualities (in comparison with each other) in terms of solving the user's tasks and meeting his/her needs.

We apply modern argumentation theory [1–9] to develop methods for argumentative dialogue between an intelligent assistant and a user.

This article is primarily devoted to the presentation of methods and technologies of selection by the intelligent assistant of the goods most suitable for the user in the process of dialogue with the user. A more detailed description of the methods of building justification and argumentation of the fact that this product is the best for the user will be the subject of the next article.

The necessary information for the dialogue with the user of the intelligent assistant is taken from the semantic (ontological) model [10–12], the structure of which we will describe below. The semantic model is filled and replenished by extracting information from the websites of product manufacturers and online shops, as well as by analyzing customer reviews of products they purchased.

In developing the ontology model, we use a number of ontologies. These are:

- ontology of the subject domain as a whole;
- ontology of characteristics, properties, functionalities of various goods and devices;
- user ontology: user tasks, goals and intentions; what goals users achieve and in what ways.

What is extremely important in the approach we developed is that we save and analyze dialogues with the user. This is a significant difference between this approach and most existing solutions.

For example, Alice (a virtual voice assistant created by Yandex, YandexGPT 2), when having a dialogue with a user, does not "remember" even the previous line or question of the user. If the user says: “Alice, put me some song by band X”, Alice will put a song by this band. But if the user says, “Alice, I like songs by band X. Put me some song of this band.”, then Alice answers: “I have nothing to answer”.

If you ask: “Alice, what is the title of this song?” while a song is playing, she will answer. If you ask: “Alice, put the previous song on”, she will. But if you ask: “Alice, what is the name of the previous song?”, she will not be able to answer.

The intelligent assistant we are developing for selecting products for the user and for generating arguments can address:

- to the entire current dialogue with this user;
- to previous dialogues with this user;
- to previous dialogues with other users.

In this way the intelligent assistant works with cases of previously conducted dialogues. They are on the third level of the four-level ontological model, which will be described in detail below.

Currently, various organizations, such as online shops, banks, etc., use virtual assistants designed to help the user find the right product or service. However, as a rule, these systems work according to a predetermined scenario and when a new situation arises that was not foreseen in advance, they are unable to assist the client and redirect him/her to interact with a human consultant.

This situation, in which the virtual assistant is unable to find a solution or give the right recommendation, decreases the user’s motivation to work with it in the future. In addition, such systems almost always conduct a dialogue from scratch, without remembering the user and the context of the dialogue. If the user has already approached a similar problem, he or she must go through the whole process of searching for a solution again.

The system we are developing automates the solution search process by reusing previously accumulated cases (situations, domain cases). By comparing the current user’s goal and information from previous dialogues, it is possible not only to find a similar solution in the past, but also to additionally argue the proposed solution based on the coincidence of intentions. This gives the developed system an advantage over existing analogues, which are incapable of accumulating cases and arguing their solutions.

In this paper, we develop a case-based approach to building an intelligent system capable of reasoning about its answers. For this purpose, we construct an argumentation graph, develop methods for structuring domain cases and use ontological homomorphisms to transform the available cases into a ready-made solution. The goal of an intelligent dialogue system is to help a person to find an answer to a particular question. Our task is to circumvent the limitation of the neural network approach, which is the inability to take into account the results of recent user sessions. For this purpose, we implement a case approach to solution construction using ontological homomorphisms. On its basis, the construction and reasoning of the solution takes place.

2. Existing approaches and solutions

Currently, there are many dialogue systems designed for different tasks: systems that support dialogue with simple phrases, voice assistants (Alice, Siri and others), capable of more complex communication, jokes or performing simple tasks (find information on the Internet, turn on an electrical appliance), etc. The top of the development of such dialogue systems are complex language universal models (LLM) designed to solve arbitrary tasks like ChatGPT.

A separate subclass of recommender systems is worth mentioning. For them it is important not only to find and output correct information, but also to justify why the system derived a certain solution. We will consider different types of systems from the point of view of approaches to solution retrieval and its justification.

2.1. ChatGPT

A dialogue system (LLM language model) from OpenAI focused on solving arbitrary tasks [13]. It is based on a strong pre-trained InstructGPT language model used to formalize user input, while the model itself is trained using the Reinforcement Learning with Human Feedback (RLHF) approach. With the help of experts, a reward model was created that assigns a score to the correctness of the solution of the underlying model, after which automated reinforcement learning was run.

There are exceptionally few research articles on the ChatGPT architecture available at this time, as OpenAI has not disclosed such information other than what is available on the company's blog [13].

The model has a number of disadvantages.

1. When having a long dialogue, the answers become unclear and the system starts to produce incorrect answers. The reason for this is that the model is not trained on long dialogues, the focus is shifted to more detailed and elaborate answers to a small number of questions in one session.
2. The initial model does not use data from the Internet, but is limited to the data that was fed into it during training. As a result, it cannot use information from dialogues with users (e.g., a new fact about the world around us), which makes the model more dependent on the quality of the training data and creates a time lag between the emergence of new knowledge and its input into the model.
3. The model does not verify the data generated, leading to a paradoxical situation in which the system reasons in great detail about meaningless things, misleading the user (this phenomenon is metaphorically called "hallucinations").

In addition, it is important to note that ChatGPT can store the context of the current dialogue, but this information will not be used in the next session with the same user. As a result, we get that the model can generate answers irrelevant for the user that have been previously received and used in a dialogue with the same user and "forget" what the user communicated to the model earlier.

At the same time, the task of pre-training a neural network of this scale, depending on the results of dialogues, to solve the problem of dynamically updating a set of domain cases requires significant computational resources and can lead to the problem of catastrophic forgetting [14], since the incoming data can be anything. As a result, developers prefer to first type new data, process it, and run a one-time but lengthy learning process.

2.2. BlenderBot

ParlAI's dialogue-oriented BlenderBot model [15]. Due to the presence of long-term memory, the system supports long dialogues better than ChatGPT. It is able to use information previously received from a user, but the data received from one user is not used in a dialogue with other users.

The architecture of the BlenderBot system is based on the pipeline principle [16]. The system generates a response by sequentially using a series of modules, each of which performs a different task, then passes its output to the next module. The model exists in three types, depending on the number of parameters (3, 30, 175 billion parameters).

The order in which the modules [16] are called depends on the context in which the dialogue takes place. The system forms a solution depending on the context of the dialogue by accessing both its own long-term memory and by forming Internet queries. In case memory and web searches are not required, the data will be retrieved from the current dialogue.

The response is generated from this data, and the modules responsible for simulating empathy and personality in the dialogue are also involved at this stage. A complete list of modules and their detailed description can be found in the BlenderBot developers' technical report [16]. By using many modules and adjusting the order in which they are invoked, the system can, while maintaining a long dialogue, update its user data and have a dialogue on several different topics, switching between them, depending on the context of the user's last message.

Considering BlenderBot in the context of our task, we can note the successful implementation in the model of a long-term memory system and a system for deciding whether to search a database of previous dialogues or the Internet. However, data about previous dialogues are stored in memory only as a set of facts (e.g., "User 1 likes dogs", "User 2 lives in country A"), have no semantic connection with each other and are tied to a specific user. Thus, most of the context of previous dialogues is lost. Also, the system cannot reproduce its own steps in solving a particular problem.

Thus, we have considered two popular language models, one of which is intended for solving arbitrary tasks, and the other for maintaining a long and complex dialogue. It should be noted that both of these models do not show the user explicitly how the solution was obtained and do not justify or argue this solution in any way.

We next consider examples of systems designed for narrower applications, but with a more structured approach to constructing argumentative dialogue.

2.3. A system of argumentative dialogue based on argumentative structures

The system presented in [17] has been designed to lead a discussion between the user and the system on various topics. It is a text-based system (although voice interface is also supported), analyses the user's messages, extracts argumentation premises from them [17], and generates arguments based on them. Programmatically, the system is implemented as a set of modules combined using Apache ActiveMQ.

The user's phrase is converted into a "dialogue action". The authors consider four types of dialogue actions: assertion, question, concession and retreat. A logistic regression-based classifier is used to recognize these actions. It then searches for a suitable argumentation node in the argumentation graph based on the cosine similarity between sentence vectors (and the similarity is considered between averaging the value of the node and the user's phrase). The extracted argumentation is processed with respect to the subject domain and the next argument is generated.

Note an important feature of this system: it is able to evaluate the user's actions, in particular, whether he continues his thought, is about to start speaking or is about to finish. This serves as additional information when generating an answer and its justification.

The argument base is populated using the automatic argument extraction techniques developed in [18]. At the time of publication of the paper [17], the system was capable of understanding five discussion topics and supporting up to 2000 argumentation nodes for each of them. However, there is no way to dynamically add new data to the argumentation structure as the dialogue progresses, so a natural question arises: how will the system react to new information that is not in its data.

2.4. Argumentation systems based on communication discourse trees

According to speech structure theory, any coherent discourse can be described by a single discourse tree, described as a tree structure using speech act theory [30]. Each paragraph of text (or the whole text) is converted into a tree through linking sentences using speech acts (e.g., "Justification"), with the leaves of the tree containing the sentences themselves. In this way, an argumentation tree is constructed from which it is possible to determine the presence of argumentation in a paragraph/text. In addition, this approach preserves the context of the argumentation, without which even a human expert would not be able to analyze the presence of argumentation.

It is important to note that in [30] it is the fact of argumentation, not its semantic part and/or persuasiveness, that is considered. Nevertheless, such an approach can be used for semantic analysis of argumentation.

The approach based on communicative discourse trees is also discussed in [31, 32].

2.5. A framework for an argumentative dialogue on the COVID-19 vaccination

The dialogue system [19] is designed to consult the user on the topic of vaccination, with maximum justification of the system recommendations. The system is based on the construction of an argumentation graph, according to the approach of Chalagin and Hunter [20]: finding out the similarity of sentences to get an answer from the knowledge base. The method does not consider the user's previous actions and, as a result, loses the context of the dialogue. The system tries to take into account the user's arguments and construct an answer that does not contradict them and, at the same time, is consistent with the knowledge base.

The reasoning module of the system [20] consists of an argumentation graph compiled by the expert. The nodes of this graph represent either state arguments or response arguments. Associated with each node is a set of natural language sentences representing possible user arguments for that node. The search for a matching node in the graph is performed using a similarity measure of the sentences. The solution is generated on the basis of the information provided by the user, and the node found should agree with the user's data and prevent "dangers," the unacceptable points of the solution indicated by the user (in the example of the article we are talking about counter-indications to vaccination). At the same time, if the system cannot find a "safe" solution, it will still issue a response to the user, but with a request for additional information to adjust the solution. Thus, each new user argument "switches on" the corresponding node in the argumentation graph, and the links coming from this node either reinforce the corresponding solution options or switch them off from the graph.

Thus, well-known universal dialogue systems are good at many tasks, but they do not have mechanisms to explain the progress of solution construction to the user. In addition, their architecture does not allow them to quickly integrate solutions from successful user sessions into their knowledge bases.

On the other hand, specialized systems, in which reasoning is an initial requirement, mainly rely on a pre-prepared knowledge base compiled by an expert and build their solutions and arguments on its basis; going beyond this knowledge base leads to the construction of an unreliable solution. This results in the inability to work with the results of previous sessions, since they lack a mechanism for inserting such information into the knowledge base.

3. Four-level ontological model of the subject domain

As stated above, the aim of this work is to create methods for an intelligent assistant (digital assistant) to conduct a reasoned dialogue with the user [21]. In the framework of our research, the development of an intelligent assistant [22] is based on a semantic model is a four-level ontological model of the subject domain [10, 11]. Let us describe this model in more detail.

The first level of the ontological model is ontologies:

1. Ontology of the subject domain of the goods (devices) under consideration is a set of concepts describing: types of devices; structure and characteristics of devices; functionality of devices.
2. User ontology is a set of concepts describing: goals and intentions; interests, desires, needs; types (classes) of tasks to be solved.
3. Dialogue ontology is a set of concepts describing: argumentation (arguments, counterarguments, etc.); emotional evaluations of users, their satisfaction or dissatisfaction; success of a given dialogue (purchase of goods by the user, continuation of communication and other goods or termination of dialogue by the user, unwillingness to continue it further).

By ontology we mean knowledge only about the meaning of concepts, i.e., analytical statements [23–25] that do not contain information about the state of the real world.

The second level of the ontological model is general (universal) knowledge. These are synthetic statements [23, 26], knowledge about the real world:

1. Subject matter theory is properties of specific goods, their characteristics, functionality, etc.
2. Knowledge about types of users, their classification (by income level, social status, educational level), classes of tasks solved by users, hierarchy of tasks, methods of reducing tasks to subtasks and the possibility of solving the same tasks with different devices.
3. Knowledge of methods of dialogues with users — area methods of argumentation, justification of specific proposals to the user; methods of identifying the goals and needs of users, tasks solved by them; methods of determining the emotional state and emotional assessments of users.

The third level of the ontological model, the most important within the framework of this paper, is the level of domain cases. These are:

1. Product and device cases are specific devices, components, accessories, device sets, price and availability of products in shops, etc.
2. User cases are those users with whom the intelligent assistant has already had dialogues, together with their properties and characteristics; knowledge about the users, their goals, intentions, interests, needs, the tasks they solve.
3. User dialogue cases, hierarchically structured: dialogue with a single user; all dialogues with a given user; dialogues with classes of users.

The fourth level of the ontological model is evaluative and probabilistic knowledge. They are generated by analyzing the domain cases contained in the third level of the ontological model. These include:

- the likelihood that a user with certain characteristics and needs will want to purchase the device;
- the probability that a user who has (bought) device **A** will want to buy device **B**;
- evaluation of similarity of domain cases: devices, device parameters, users and dialogues with them.

Based on this four-level semantic model, we develop a precedence-based approach to construct a reasoned dialogue between an intelligent assistant and a user.

The neural network algorithms that are actively used now, due to their structure, are limited in using recent cases in their model; only when training the next version can this data be included in the training sample.

The use of the case approach solves this problem: we can add new precedents "on the fly," while the system is running. The use of the precedent approach also makes the dialogue system capable of arguing its own conclusions, justifying the choice of goods offered to the user.

The case approach relies on a set of case examples from past user sessions. For certain subject domains, it allows for building solutions from existing data by applying some transformations to it, changing the structure of the case-solution according to the user's task (such transformations, in particular, are realized by means of ontological homomorphisms of partial models, which will be described in detail below).

When implementing the case approach, a number of problems arise. First, precedents should be structured, and not as a relational table with a set of columns. This way of organization will make the search for a suitable precedent weakly related to the semantic content of the precedent. Secondly, it is necessary to evaluate the degree of similarity of precedents, both for finding a suitable "starting point" and for converting a precedent into a final decision. Thirdly, the system should be able to transform precedents according to the user's requirements.

The solution to these problems is to organize precedents into a semantic graph, where the links between precedents will reflect their similarities, showing the degree of similarity in one or another property of the precedent. This solves the problem of semantic search. We need to traverse the precedent graph following the desired semantic links. In this case, the distance of two nodes from each other will explicitly reflect the degree of similarity of the corresponding precedents. The process of precedent transformation can be considered as a transformation of partial models formally describing these precedents by means of ontological homomorphisms, extensions and contractions of partial models. In this case, the properties and parameters of the original precedent will be transformed not necessarily into the same concepts, but into ontologically similar ones. Isomorphic embedding is also possible: expansion of the original precedent, as well as contraction of the precedent, removal of unnecessary elements of the model.

Thus, we implement a software system that performs solution search based on semantic similarity of precedents with an explainable solution search mechanism.

4. A theoretical and modelling approach to the design of an intelligent assistant

It is important to note that most of the precedents we consider, both product precedents and user precedents, contain only a part of all the information about the user or device. Therefore, within the model-theoretic approach, precedents should be formally described by partial models rather than by ordinary models (algebraic systems).

Definition. Consider a signature $\sigma = \langle P_1, ..., P_m, c_1, ..., \mathrm{c}_l \rangle$ in which $P_1, ..., P_m$ are predicate symbols and $c_1, ..., \mathrm{c}_l$ are symbols of constants. Consider a tuple $\mathfrak{A}^p = \langle A, P_1, ..., P_m, c_1, ..., \mathrm{c}_l \rangle$ and let for each $n \leq m$ value *n-ary* predicate P_i on $\mathfrak{A}^p$ is defined as a pair $P_i^{\mathfrak{A}^p} = (P_i^+, P_i^-)$ where $P_i^+, P_i^- \subseteq \left|\mathfrak{A}^p\right|^n$ and $P_i^+ \cap P_i^- = \varnothing$. Let us call $\mathfrak{A}^p$ a partial model in the signature σ. Let us assume that for elements $a_1, ..., a_n \in |\mathfrak{A}^p|$ if $(a_1, ..., a_n) \in P_i^+$, then it is fulfilled $\mathfrak{A}^p \vDash P_i(a_1, ..., a_n)$, if $(a_1, ..., a_n) \in P_i^-$, then it is fulfilled $\mathfrak{A}^p \vDash \neg P_i(a_1, ..., a_n)$, and if $(a_1, ..., a_n) \notin (P_i^+ \cup P_i^-)$, then the value of the predicate $P_i(a_1, ..., a_n)$ on the partial model $\mathfrak{A}^p$ is undefined.

The class of partial signature models σ denote by $K^p(\sigma)$.

We use ontological homomorphisms to transform partial models that formalize precedents. In this paper, we consider three types of ontological homomorphisms that are most important for this presentation; to illustrate, we take the example of a device like a laptop. These are ***generalization*** homomorphisms (in the partial model, the presence of a *USB A* connector in the laptop is replaced by just the presence of a *USB* connector), ***refinement homomorphisms*** (the presence of a *USB connector* in the laptop is replaced by the presence of *USB A*) and ***similarity*** homomorphisms (the presence of a *USB A connector is* replaced by the presence of a *USB C connector*).

Let us give strict definitions of ontological homomorphisms. For this purpose, let us consider ontological relations: on the set of key concepts of the ontology of the subject domain of the goods (devices) under consideration is signature predicates σ. We introduce two two-place relations: the general-private relation ***Hyp***(Q, P) and similarity relation ***Sim***(P, Q). The relation ***Hyp*** is a partial order, and the relation ***Sim*** is reflexive and symmetric (but not necessarily transitive).

Definition. Consider partial models $\mathfrak{A}^p \in K^p(\sigma_1)$ and $\mathfrak{B}^p \in K^p(\sigma_2)$ let $P^n \in \sigma_1$, $Q^n \in \sigma_2$, $\sigma_1 \backslash \{P^n\} \subseteq \sigma_2$ and it is fulfilled ***Sim***(P, Q). Mapping $h{:}|\mathfrak{A}^p| \to |\mathfrak{B}^p|$ let us call the ontological homomorphism of the **similarity of the** partial model $\mathfrak{A}^p$ into a partial model $\mathfrak{B}^p$ $(h{:}\,\mathfrak{A}^p \to \mathfrak{B}^p)$ if for any $c \in \sigma_1$ and $a_1, ..., a_n \in |\mathfrak{A}^p|$ is fulfilled:

(a) if $\mathfrak{A}^p \vDash P(a_1, ..., a_n)$ then $\mathfrak{B}^p \vDash Q(h(a_1), ..., h(a_n))$;

(b) if $\mathfrak{A}^p \vDash \neg P(a_1, ..., a_n)$ then $\mathfrak{B}^p \vDash \neg Q(h(a_1), ..., h(a_n))$;

(c) $h(c^{\mathfrak{A}^p}) = c^{\mathfrak{B}^p}$.

The truth and falsity of the other predicates from the σ_1 are preserved.

Definition. Consider partial models $\mathfrak{A}^p \in K^p(\sigma_1)$ and $\mathfrak{B}^p \in K^p(\sigma_2)$, $\sigma_1 \backslash \{P^n\} \subseteq \sigma_2$ let $P^n \in \sigma_1$, $Q^n \in \sigma_2$, $\sigma_1 \backslash \{P^n\} \subseteq \sigma_2$ and it is fulfilled ***Hyp***(Q, P). Mapping $h{:}|\mathfrak{A}^p| \to |\mathfrak{B}^p|$ let us call the ontological homomorphism of the ***generalization of the*** partial model $\mathfrak{A}^p$ into a partial model $\mathfrak{B}^p(h{:}\,\mathfrak{A}^p \to \mathfrak{B}^p)$ if for any $c \in \sigma_1$ and $a_1, ..., a_n \in |\mathfrak{A}^p|$ is fulfilled:

(a) if $\mathfrak{A}^p \vDash P(a_1, ..., a_n)$ then $\mathfrak{B}^p \vDash Q(h(a_1), ..., h(a_n))$;

(b) $h(c^{\mathfrak{A}^p}) = c^{\mathfrak{B}^p}$.

The truth and falsity of the other predicates from the σ_1 are preserved.

Definition. Consider partial models $\mathfrak{A}^p \in K^p(\sigma_1)$ and $\mathfrak{B}^p \in K^p(\sigma_2)$ let $^n \in \sigma_1$, $Q^n \in \sigma_2$, $\sigma_1 \backslash \{P^n\} \subseteq \sigma_2$ and it is fulfilled ***Hyp***(Q, P). Mapping $h{:}|\mathfrak{A}^p| \to |\mathfrak{B}^p|$ let us call the ontological homomorphism of the ***refinement of the*** partial model $\mathfrak{A}^p$ into the partial model $\mathfrak{B}^p$ $(h{:}\,\mathfrak{A}^p \to \mathfrak{B}^p)$ if for any $c \in \sigma_1$ and $a_1, ..., a_n \in |\mathfrak{A}^p|$ is fulfilled:

(a) if $\mathfrak{A}^p \vDash \neg P(a_1, ..., a_n)$, then $\mathfrak{B}^p \vDash \neg Q(h(a_1), ..., h(a_n))$;

(b) $h(c^{\mathfrak{A}^p}) = c^{\mathfrak{B}^p}$.

The truth and falsity of the other predicates from the σ_1 are preserved.

The single or multiple use of ontological homomorphisms allows the intelligent assistant to automatically switch from descriptions of some devices to descriptions of other devices that are similar to a certain extent. For example, a user wants to buy a certain device with certain characteristics, but the required device is not available (or its price does not suit the user). Then the intelligent assistant automatically finds another device, whose partial model is ontologically homomorphic to the model of the original one, but which is available for sale, and offers this device to the user. The product (or several products) closest to the required one is automatically searched.

When suggesting devices to the user, the intelligent assistant also provides an explanation of why their difference from the user's desired one is not essential from the point of view of the tasks to be solved by the user. Such natural language explanations are either pre-defined in the semantic model, when determining the ontological similarity of concepts, or extracted from natural language texts in the process of dialogue (in particular, from product descriptions on the websites of manufacturers and shops, from customer reviews, etc.). [27, 28]).

This process continues iteratively until a device satisfactory to the user is found: the user indicates what he does not like, and the intelligent assistant selects a new variant. In this way, a reasoning graph is constructed whose vertices contain partial models corresponding to the devices, and transitions are made using ontological homomorphisms.

Here it is important to note that by precedents we mean all kinds of objects, subjects and situations with which the intelligent assistant works. The precedents are both objects of the subject domain, users themselves, and dialogues with users, the results of the previous session of the software system: artifacts of interaction between the intelligent assistant and the user. As an example, let us consider a hierarchy of structured precedents: objects of the subject domain related to the satisfaction of user needs:

1. A subject matter object, a commodity that a user needs (e.g., a computer or smartphone).

2. Subject matter object + user needs (as identified by the intelligent assistant in the dialogue process).

3. Subject matter object + user needs + class of tasks to be solved by the user. We extend the precedent by adding the tasks that the user needs to solve. It is important to note that the properties of the subject matter object (e.g., the functionality of the device) are clearly defined in the knowledge base and are independent of the user's perception. On the other hand, the tasks that the user intends to solve with a given device depend on the user's ultimate goals, desires and needs. The class of tasks to be solved is determined by the user. Having data about the user's needs and the tasks to be solved by the user, we construct a precedent as a triple: <partial model describing the device; user's needs; set (class) of tasks to be solved by the user>.

This way of representing precedents facilitates combining objective information about goods with subjective information about the user obtained by the intelligent assistant in the process of dialogue. Recall that precedents are represented at the third level of the ontological model of the subject domain.

The construction of a solution is the selection of goods required by the user and involves the comparison of both objects of the subject domain (goods, devices) and structured precedents described above. For this purpose, the apparatus of metrics is used. This will allow comparison of objects and precedents during the work of the intelligent assistant. In particular, a semantic graph of precedents with a precalculated (or set by an expert) measure of similarity (likeness) of precedents is used. Knowledge about similarity measures of precedents belongs to the fourth level of the ontological model of the subject domain. On the basis of this knowledge, in particular, the similarity relation is specified The similarity relation discussed above is in the definition of ontological homomorphisms.

When selecting products, user priorities are calculated based on two parameters: firstly, properties and functionality of devices and, secondly, user's needs and desires, class of tasks to be solved, that is, objective and subjective parameters. Focusing on these two types of parameters, we calculate the similarity of different precedents, including the objects of a given subject domain.

5. Software implementation of the dialogue system

The developed software system [22] is a set of five blocks (modules) that provide various stages of the system operation. The technical implementation is a MVC application on Java Spring, with REST interface.

Block 1 is responsible for performing user input and formalizing it through a speech action search mechanism [29], correcting the user model and detecting user intentions.

Block 2 is responsible for analyzing the input received, and generating system messages to request further information from the user.

Block 3 is responsible for searching for the required product, a precedent of the subject domain. It checks whether the partial model of the required product, built as a result of the dialogue with the user, is a submodel of the model of some precedent of the subject domain (i.e., the product available). The input of the block is the user's needs formulated and checked for consistency. Then, with the help of ontological homomorphisms realized on the basis of the similarity function of two partial models, the precedent most similar to the desired user is searched for.

If a precedent is found, this solution will be proposed to the user; the natural language description of the found precedent will be used as a justification (argument).

Block 4 is responsible for analyzing the user's reaction. The main function of the block is to clarify the user's requirements. If there are new data in the user's response, they are formalized using the mechanisms of blocks 1 and 2 and block 3 is started again. Thus, the solution search process is iterative.

Block 5 is responsible for the final generation of the decision and its justification (reasoning). The user is offered a product that fully meets his requirements identified in the dialogue process. A set of goods that meet the requirements but differ in price or characteristics that are not important for the user can also be presented.

Conclusion

This article develops methods of creating intelligent assistants. Intelligent assistants can be used to help users choose products as recommendation systems in call centers to solve various customer problems, to solve technical support tasks, to help people with disabilities. In this paper, first of all, we consider intelligent assistants designed to help the user to select goods.

To create intelligent assistants, we develop methods for reasoned dialogue with the user. For this purpose, we develop methods for automated construction of reasoning and argumentation. The formalization of reasoning and argumentation is done using partial models, homomorphisms and ontological homomorphisms of partial models. Ontological homomorphisms of the similarity of partial models formally describe the similarity of precedents, which serves as a mathematical basis for the construction of reasoning based on precedents.

The proposed architecture of the software system implements methods of dialogue with the use of precedents, replenishment of the database of precedents after each session of work, their organization in the form of a semantic network. Such an approach allows us to achieve transparency of the system operation, to increase flexibility of solution selection due to the analysis of semantic content of phrases entered by the user (with the help of atomic models), all of which distinguishes the system from the existing analogues.

Further development of the system is possible in the direction of improving the algorithm of precedent search in the semantic network, with the introduction of more links between precedents to increase the detail of the search, as well as the development of methods for determining the similarity and likeness of precedents. ■

Acknowledgements

The work was carried out within the framework of the state assignment of the S.L. Sobolev Institute of Mathematics, Siberian Branch of the Russian Academy of Sciences. S.L. Sobolev Institute of Mathematics, Siberian Branch of the Russian Academy of Sciences, project FWNF-2022-0011.

References

1. Toulmin S.E. (2003) *The uses of argument*. Cambridge University Press.

2. Reed C., Rowe G. (2004) Araucaria: Software for argument analysis, diagramming and representation. *International Journal on Artificial Intelligence Tools*, vol. 13, no. 4, pp. 961–979.

3. Walton D., Reed C., Macagno F. (2008) *Argumentation schemes*. Cambridge University Press.

4. Macagno F., Walton D. (2015) Classifying the patterns of natural arguments. *Philosophy & Rhetoric*, vol. 48, no. 1, pp. 26–53. https://doi.org/10.5325/philrhet.48.1.0026

5. Wagemans J. (2016) Constructing a periodic table of arguments. *Argumentation, objectivity, and bias: Proceedings of the 11th international conference of the Ontario Society for the Study of Argumentation, Windsor, Ontario, Canada*, pp. 1–12. https://doi.org/10.2139/ssrn.2769833

6. Walton D., Gordon T.F. (2017) Argument invention with the Carneades argumentation system. *SCRIPTed*, vol. 14, no. 2, pp. 168–207. https://doi.org/10.2966/scrip.140217.168

7. Wachsmuth H., Potthast M., Al-Khatib K., Ajjour Y. (2017) Building an argument search engine for the Web. Proceedings of the *4th Workshop on Argument Mining*, pp. 49–59.

8. Slonim N. et al. (2021) An autonomous debating system. *Nature*, vol. 591, no. 7850, pp. 379–384. https://doi.org/10.1038/s41586-021-03215-w

9. Stab C., Miller T., Shiller B., Rai P., Gurevych I. (2018) Cross-topic argument mining from heterogeneous sources. Proceedings of the *2018 Conference on Empirical Methods in Natural Language Processing*, pp. 3664–3674.

10. Naydanov Ch., Palchunov D., Sazonova P. (2015) Development of automated methods for the critical condition risk prevention, based on the analysis of the knowledge obtained from patient medical records. Proceedings of the *International Conference on Biomedical Engineering and Computational Technologies, Novosibirsk, 28–30 October 2015*, pp. 33–38. https://doi.org/10.1109/SIBIRCON.2015.7361845

11. Naydanov Ch., Palchunov D., Sazonova P. (2016) Development of automated methods for the prevention of risks of critical conditions, based on the analysis of the knowledge extracted from the medical histories. *The Siberian Scientific Medical Journal*, vol. 36, no. 1, pp. 105–113.

12. Naidanov Ch.A. (2019) Development of ontological model kernel, customisable for the subject domain. *Vestnik of Novosibirsk State University. Series: Information technologies*, vol. 17, no. 1, pp. 72–81.

13. OpenAI (2022) *Introducing ChatGPT*. Available at: https://openai.com/blog/chatgpt (accessed 26 January 2024).

14. McCloskey M., Cohen N.J. (1989) Catastrophic interference in connectionist networks: The sequential learning problem. *Psychology of Learning and Motivation*, vol. 24, pp. 109–165.

15. Blenderbot AI (2022) *BlenderBot 3: A Conversational AI Prototype*. Available at: https://blenderbot.ai (accessed 26 January 2024).

16. Shuster K. et al. (2022) Blenderbot 3: A deployed conversational agent that continually learns to responsibly engage. *arXiv:2208.03188*. https://doi.org/10.48550/arXiv.2208.03188

17. Higashinaka R. et al. (2017) Argumentative dialogue system based on argumentation structures. Proceedings of the *21st Workshop on the Semantics and Pragmatics of Dialogue*, pp. 154–155.

18. Lippi M., Torroni P. (2016) Argumentation mining: State of the art and emerging trends. *ACM Transactions on Internet Technology*, vol. 16, no. 2, pp. 1–25. https://doi.org/10.1145/2850417

19. Fazzinga B., Galassi A., Torroni P. (2021) An argumentative dialogue system for COVID-19 vaccine information. *International Conference on Logic and Argumentation*. Cham: Springer International Publishing, pp. 477–485.

20. Chalaguine L.A., Hunter A. (2020) A persuasive chatbot using a crowd-sourced argument graph and concerns. *IOS Press E-books Series Frontiers in Artificial Intelligence and Applications*, vol. 326: Computational Models of Argument, pp. 9–20. https://doi.org/10.3233/FAIA200487

21. Palchunov D., Yakobson A. (2022) Automated methods for conducting an argumented dialogue with the user. *2022 IEEE International Multi-Conference on Engineering, Computer and Information Sciences (SIBIRCON)*, pp. 880–885. https://doi.org/10.1109/SIBIRCON56155.2022.10016975

22. Palchunov D.E., Yakobson A.A. (2022) *Software system "ArgumentDialogue"*. Certificate of state registration of computer programme No. 2022664041, registered 22.07.2022.

23. Carnap R. (1956) *Meaning and necessity. A study in semantics and modal logic*. Chicago.

24. Palchunov D.E. (2006) Modelling of thinking and formalization of reflexion I: Theoretical-model formalization of ontology and reflexion. *Philosophy of Science*, vol. 4(31), pp. 86–114.

25. Palchunov D.E. (2022) Methodological aspects of the application of model theory in knowledge engineering and artificial intelligence. *2022 Ural-Siberian Conference on Computational Technologies in Cognitive Science, Genomics and Biomedicine (CSGB)*, pp. 210–215. https://doi.org/10.1109/CSGB56354.2022.9865602

26. Palchunov D.E. (2008) Modelling of thinking and formalization of reflexion. Part 2: Ontologies and formalisation of concepts. *Philosophy of Science*, vol. 2(37), pp. 62–99.

27. Palchunov D.E. (2008) Solution of the information search task on the basis of ontologies. *Business-informatics*, vol. 1(3), pp. 3–13 (in Russian).

28. Orlovsky A., Palchunov D. (2021) Development of automated methods for the domain ontology population with the help of a virtual assistant. *2021 IEEE 22nd International Conference of Young Professionals in Electron Devices and Materials (EDM)*, pp. 537–541. https://doi.org/10.1109/EDM52169.2021.9507641

29. Makhina E.D., Palchunov D.E. (2018) Software system to determine speech actions in natural language texts. *Vestnik of Novosibirsk State University. Series: Information technologies*, vol. 16, no. 4, pp. 95–106. https://doi.org/10.25205/1818-7900-2018-16-4-95-106

30. Galitsky B., Ilvovsky D., Kuznetsov S. (2018) Detecting logical argumentation in text via communicative discourse tree. *Journal of Experimental and Theoretical Artificial Intelligence*, vol. 30, no. 5, pp. 637–663. https://doi.org/10.1080/0952813X.2018.1467492

31. Galitsky B., Ilvovsky D., Kuznetsov S.O. (2015) Text integrity assessment: Sentiment profile vs rhetoric structure. *Computational Linguistics and Intelligent Text Processing* (eds. A. Gelbukh). CICLing 2015. Lecture Notes in Computer Science, vol. 9042. Springer, Cham, pp. 126–139. https://doi.org/10.1007/978-3-319-18117-2_10

32. Galitsky B., Ilvovsky D., Kuznetsov S.O. (2015) Rhetoric map of an answer to compound queries. Proceedings of the *53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, Beijing, China*, vol. 2: Short Papers, pp. 681–686. https://doi.org/10.3115/v1/P15-2112

About the authors

Dmitry E. Palchunov

Dr. Sci. (Phys.-Math.), Academician of the Russian Academy of Engineering;

Leading Researcher, S.L. Sobolev Institute of Mathematics, Siberian Branch of the Russian Academy of Sciences, 4, Akademika Koptyuga Ave., Novosibirsk 630090, Russia;

E-mail: palch@math.nsc.ru

ORCID: 0000-0001-9487-3256

Alexander A. Yakobson

Assistant, Department of General Informatics, Novosibirsk National Research State University, 1, Pirogova St., Novosibirsk 630090, Russia;

E-mail: a.yakobson@g.nsu.ru