On a Chatbot Navigating a User through a Concept-Based Knowledge Model

Elizaveta Goncharova
National Research University
Higher School of Economics
Moscow, Russia
egoncharova@hse.ru

Dmitry Ilvovsky
National Research University
Higher School of Economics
Moscow, Russia
dilvovsky@hse.ru

Boris Galitsky
Oracle Inc.
Redwood Shores, CA, USA
boris.galitsky@
oracle.com

Abstract

Information retrieval chatbots are widely used as assistants, to help users formulate their requirements about the products they want to purchase, and navigate to the set of items that satisfies their requirements in the best way. The work of the modern chatbots is based mostly on the deep learning theory behind the knowledge model that can improve the performance of the system. In our work, we are developing a concept-based knowledge model that encapsulates objects and their common descriptions. The leveraging of the concept-based knowledge model allows the system to refine the initial users' requests and lead them to the set of objects with the maximal variability of parameters that matters less to them. Introducing the additional textual characteristics allows users to formulate their initial query as a phrase in natural language, rather than as some standard request in the form of, "Attribute - value".

1 Introduction

In recent years, a number of chatbots for products and services exploration has been proposed (Ukpabi and Karjaluoto, 2019; Gao et al., 2019).

Chatbots are noticeably one of the most popular AI technologies across the world. From simplifying business workflows, enhancing employee and customer experience to reducing costs, chatbots provide numerous benefits for organizations of all sizes. Besides Customer Service, chatbots support sales, marketing, IT Helpdesk, business intelligence, Intranet, and HR. The use cases of chatbots are diverse and vary across departments and industries. Most enterprise leaders are yet to understand and explore the full potential of chatbots. In spite of the efforts spent on scientific philosophy of chatbots, most are yet to produce systems capable of an efficient deployment in real-world environments. These efforts are mostly spent on a learning theory behind the conception of the chatbots. Today's level of scientific knowledge is way ahead of the real-world deployment of chatbots.

Dialogue management is one of the bottlenecks of a successful chatbot, and regretfully each of the above domains requires a specialized dialogue structure. Some dialogue structures are covered by discourse representations of the text being communicated. An exploration of products with features and attributes requires a distinct navigational structure such as lattice (a partially ordered set of pairs of product lists and features they share). In (Makhalova et al., 2019) the authors have explored how a user can walk along a lattice picking a product he needs. Once products and their features are available, we build a lattice and navigate it. However, features of products are not always available in explicit way of direct association, in a form of a database: frequently, they need to be extracted from text.

In this paper, we extract associations between products and features from text. To do that, we first reveal product names as entities, together with their features and attributes, from a corpus of documents. Then the noun phrases for products are aggregated, grouped, filtered, and cleaned. Also, these noun phrases need to occur in opinionated expressions. Moreover, we conduct an argumentation analysis giving a preference to product features being the subject of argumentative expressions of the users in their reviews.

Proceedings of the Workshop on Natural Language Processing in E-Commerce (EComNLP), pages 53–65
Barcelona, Spain (Online), Dec 12, 2020.

Opinion mining or sentiment analysis detects customers' opinions, sentiments, emotions, appraisals, and attitudes towards products and services. Chatbots employ NLP and sentiment analysis for multiple purposes including human-machine communication applied to business, education, and health.

The paper is organized as follows. In Section 2 we present the basic description of the proposed model. In Section 3 we overview the algorithm for textual data preparation. In Section 4 we give a brief overview of the proposed knowledge-based model. In the 5th section we describe the possible scenarios for navigation through the model. In section 6 we provide a few examples of the dialogue conducted by the proposed chatbot model. We conclude and give the directions for future work in Section 7.

2 Model description

Information retrieval (IR) chatbots represent systems that provide web search in case of imprecise queries in specific domains. In many cases the user has only general idea about the product he wants to purchase, so, the goal of IR chatbots is to help the user to find the desired item, and simultaneously to refine his initial idea about the product.

While the work of the standard IR chatbots is based on sending simple queries to the database and requiring the user to refine the set of predefined standard, or catalog, features (the price, e.g.), knowledge-based IR models have proved their effectiveness w.r.t. to the ability to interactively refine the query. For example, (Makhalova et al., 2019) proposed lattice navigation chatbot model that incorporates a concept-based knowledge model and an index-guided traversal through it to ensure the discovery of information relevant for users and coherent to their preferences. This chatbot not only supports a search session, but also helps users to discover properties of items and sequentially refine an imprecise query. However, it enables the chatbot to process only the standard features, and ignores text descriptions of an object, which could allow the users to grasp some additional characteristics of the product he wants to buy. In our work we propose to enlarge the proposed in (Makhalova et al., 2019) knowledge model with the additional characteristics, extracted from textual descriptions of the products.

The online stores usually contain a database of digital reviews, where a person may refer to find some additional information about the product. If we encapsulate this kind of information into the knowledge model, the system will be able to provide more additional characteristics to the user, while the user can formulate his requests not just in the "attribute - value" form, but also as the phrase in natural language.

The pipeline shown in Figure 1 illustrates the scheme of the proposed IR system. It combines several blocks of data preparation responsible for retrieving the informative features from the text and enlarging the catalog data, building the lattice-based knowledge model, and navigation through the lattice with the chatbot to retrieve the information relevant for the user.

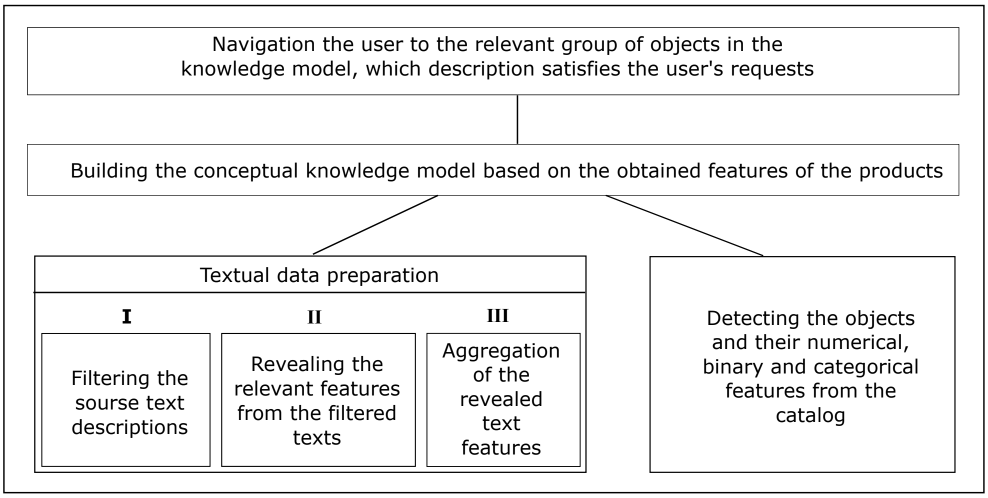


Figure 1: The proposed pipeline includes three main blocks: preparation of the text data, building the knowledge model, and navigation through the model

3 Textual data preparation

Let us overview the pipeline that illustrates the main blocks of the proposed IR system. The first block is the preparation of textual data that we have divided into three parts. The first part is responsible for filtering the initial text descriptions, which is especially important when the source of information is the users' feedbacks. We would like to retrieve text features only from the useful, or reasoned reviews, hence, we need to filter the initial texts by their validity. In the second step, the system reveals informative parts from the filtered text descriptions. On the one hand, this part of the pipeline allows the system to reduce the size of processed data and, on the other hand, to process only those parts of texts that refer to the object characteristics. We perform this part via parsing the initial paragraph of the text and retrieving the noun phrases as the most common descriptors of the object. The final step is responsible for aggregation of the text features. Some noun phrases occur only in several reviews, which means that this description of an object is not supported by many users, so we claim that it is a plain biased opinion, and the system does not use these descriptors. We propose to use only the features that are revealed from 30% percent of the reviews. In the following subsections, we will describe each part of the first block in more detail.

3.1 Filtering text data

At this stage, the system identifies useful reviews in the initial dataset. We assume that useful reviews that we should consider during the follow-up procedure of text features retrieval should include some argumentation part rather than just sentiment or plain opinion about the product. If we filter out the useless reviews, the system will gather informative text features only from the feedbacks that contain some reasons or evidence of why the product is good or bad. Argument mining techniques are able to capture these specificity of the text. So, we propose to classify the reviews w.r.t. to their argumentative power.

Argument mining itself is a challenging NLP task, nowadays the researchers find different ways to solve it. In our project, we investigate the influence of various features on the results of argumentation mining. We build a binary classifier that divides the feedbacks into the argumentative and non-argumentative ones. The classifier is based on simple linguistic features, such as tf-idf or bag of words, and some additional features corresponding to the argumentation, which are believed to improve the classification performance.

The one source of argumentation features that we utilize in our discriminative model is the automatic MARGOT tool, which is an online argumentation mining system. The MARGOT provides the segmentation of the text on claim and premise parts and reveals several scores that outline the power of claim and evidence in the sentence. Another additional source of information is discourse. There are research (Musi et al., 2018; Hewett et al., 2019) that prove the strong correlation between the rhetoric relation and argumentation relations in the texts. We use state-of-the-art discourse parser that can detect the discourse connections among the text parts, and argue that if we enlarge the linguistic features of the text described above with some discourse information, we can also improve the performance of our classifier. In further subsections we briefly describe these two additional argumentation characteristics.

3.1.1 Argumentation filtering

We use MARGOT system as a source of additional features that could improve the performance of detection argumentative reviews. MARGOT is publicly available argumentation mining system that performs two tasks, the former is segmentation of text into claim and premise, and the latter is the detection of claim and evidence scores for each sentence in the input text. These scores are used to assess the strength of claim and evidence. We propose to encapsulate these scores as the additional characteristics that will be used by our classifier for argument presence detection.

In (Passon et al., 2018), the authors propose to calculate the statistics of how many sentences in the review were defined by MARGOT as argumentative and use this statistics as the additional features. In our research we do not calculate such kind of information, however, we compute two additional characteristics for the claim and evidence scores, respectively. These features are calculated as the maximum claim score detected for each sentence, and the maximum evidence score, which is also detected for each sentence in the text. These characteristics simply define if the text contains at least one argument component or not: $score_{cl} = max_{s \in S}(score_{claim})$, $score_{ev} = max_{s \in S}(score_{evidence})$, where S is a set of sentences compiling the review, $score_{evidence}$ ($score_{claim}$) is the evidence (claim) score calculated by the MARGOT system for these sentences of the review. So, this additional information enlarges our data and improves the performance of the argument classifier.

The examples below show the output of the MARGOT system for two reviews and the $score_{cl}$ and $score_{ev}$ calculated for these reviews.

R_1: In my opinion, the 2020 MacBook Air is a perfect laptop. It has physical keys that are a pleasure to type on, a beautiful screen, an optimal size, good pricing, and a quad-core processor.

R_2: Simply, DON'T buy it!

	MARGOT output		Calculated features	
Text	$score_{evidence}$	$score_{claim}$	$score_{ev}$	$score_{cl}$
In my opinion , the 2020 MacBook Air is a perfect laptop.	0.11	-0.96	0.11	-0.27
It has physical keys that are a pleasure to type on, a beautiful screen, an optimal size, good pricing, and a quad-core processor.	-0.12	-0.27		
Simply, DON'T buy it!	No argument components have been found		None	None

So, the first review was defined by the MARGOT as argumentative, as its maximum evidence score is above zero. Clearly, this is a useful review. The reviewer expresses his opinion about a product and specifies its parameters that have motivated his satisfaction with the product. While the author of the second review expressed just his plain opinion about the product and did not explain it in any way. There the MARGOT system outputs that there is no argumentation in the text, and we omit this text from further processing.

3.1.2 Discourse features

As another source of additional information that could be useful in detecting the argumented review, we utilize the discourse features. During recent years the researchers have revealed a strong correlation between argumentation and discourse relations (Mann and Thompson, 1988) that could be retrieved from the rhetoric structure of the text. The investigation in this sphere claims that the discourse features can represent the corresponding argument relation (Galitsky et al., 2018). For example, the argumentation relation detail corresponds to elaboration relation in RST, while antithesis in rhetoric structure theory (RST) could correspond to the attack argument. Thus, when the one deals with the task of argument mining it seems reasonable to use information obtained from the RST, and utilize the discourse features in order to find some argument structure or simply detect the existence of argumentative relation in the text. Several effective discourse parsers could be used for making the automatic rhetoric parsing of the input text (Ji and Eisenstein, 2014; Lin and Kan, 2014).

We derive features such as elaboration, purpose, antithesis, circumstance, etc. from the discourse tree obtained with (Ji and Eisenstein, 2014) parser. We treat them as categorical features and encode using one-hot encoding. Further, these characteristics enlarge the input data and train an XGBoost classifier (Chen and Guestrin, 2016) to predict whether an argument relation exists.

The training data used for our research is obtained from the public Amazon Reviews dataset (McAuley and Leskovec, 2013). This dataset contains the users' evaluation of some product, text of the review, and metadata. Metadata contains the rating of the review (whether it was judged by other users as useful, or not). We split all the reviews into two classes in accordance with their rating of usefulness. If at least 70% of the people who voted the review judged it as useful, the review goes to the positive class, in another case — to the negative. Then we train the classifier to detect the useful and not useful reviews.

3.2 Revealing relevant textual features

After the system cleaned the data by filtering out useless reviews it proceeds to retrieve relevant features from these filtered texts. By relevant features, we mean the parts of the text which correspond to the description of a product. For example, in the sentence *"I think this is a good laptop"* only the phrase *"good laptop"* is referred to as an object description, while other words are irrelevant for our task and should be omitted.

We believe that noun phrases are the main descriptors of an object. For example, *[NP (a beautiful screen)*, *[NP (good laptop)]*. To identify such object descriptors we parse the sentences of the reviews and construct the parse trees, then we reveal only noun phrases and keep them as the description of some specific product.

This procedure refers to the second stage of data preparation. Obtained features are believed to express some characteristics of the product, however, if the descriptors were revealed from one or two reviews out of hundreds, it is rather a biased opinion of some particular reviewer, than the objective one. So as the final stage of data preparation we propose to aggregate the reviews into the groups and use only those descriptors that are supported by some predefined percentage of the reviewers.

3.3 Data aggregation

Data aggregation allows the system to combine noun phrases into groups and use the phrase as a feature to describe an object, only if it has occurred in some percentage of useful reviews. To provide this we calculate pairs consisting of the reviews and their common descriptions in the form of the noun phrases (Strok et al., 2014). These pairs are called concepts, and the order defined for them makes an algebraic lattice, called concept lattice (Galitsky et al., 2013).

3.3.1 Knowledge-based model

To construct the lattice and aggregate the descriptors (the noun phrases) we apply pattern structures (Ganter and Kuznetsov, 2001) that is defined as a triple $(G, (D, \sqcap), \delta)$, where G is a set of objects, $(D, \sqcap)$ is a complete meet-semilattice of descriptions and $\delta \ \rightarrow D$ is mapping an object to a description. The Galois connection between set of objects and their descriptions is also defined as follows $A^{\square} = \sqcap_{g \in A} \delta\left(g\right)$, $d^{\square} = \{g \in G | d \sqsubseteq \delta\left(g\right)\}$ for $A \ \subseteq G$, for $d \in D$.

A pair $< A, d >$ for which $A^{\square} = d$ and $d^{\square} = A$ is called a pattern concept. So, if we treat A as the set of texts describing each product and d as their common description, we can calculate the concept lattice. At each level of the lattice, the concepts provide different degrees of data aggregation. Starting from the most specific concepts $< A_1, d_1 >$ at the first level of the lattice, where A_1 consists of just one review, and d_1 is the set of all possible noun phrases constituting the review from A_1, and ending with the final concept $< A_{last}, d_{last} >$, where A_{last} is the set of all reviews about some particular product, and d_{last} is a set of noun phrases that are common for all the reviews.

To calculate the concepts we imply the following steps:

- Consider the set of texts (users reviews on some specific product) R.
- For each review $r_i \in R$ calculate the set of their noun phrases $\delta(r_i)$.
- Build pattern structure for the reviews R and their descriptions applying standard algorithm (AddIntent (van der Merwe et al., 2004) or CbO (Kuznetsov, 1993)). The intersection operation $\sqcap$ for descriptions is defined as follows, parts of speech tags are strictly match, while the word vertices themselves are labeled by wildcards (*).

Let us consider a small example of four users' reviews about some laptop from the Amazon web site and build a lattice with the algorithm proposed above.

R_1: *In my opinion, the 2020 MacBook Air is a perfect laptop. It has physical keys that are a pleasure to type on, a beautiful screen, an optimal size, good pricing, and a quad-core processor.*

R_2: *The MacBook Air is an extremely slow laptop. Right out the box it was slow and even after updating it is still slow and seems to only be getting slower day by day.*

R_3: *The new magic keys are amazing when typing. Unlike the 2019 MacBook Air. It's so amazing how you pay $999 for double the storage you got form last year that's starts at $1099. It is very fast and smooth.*

R_4: *The 2020 MacBook Air is a perfect laptop due to its good pricing.*

In Figure 2 there is a diagram of pattern structure lattice constructed for these reviews.

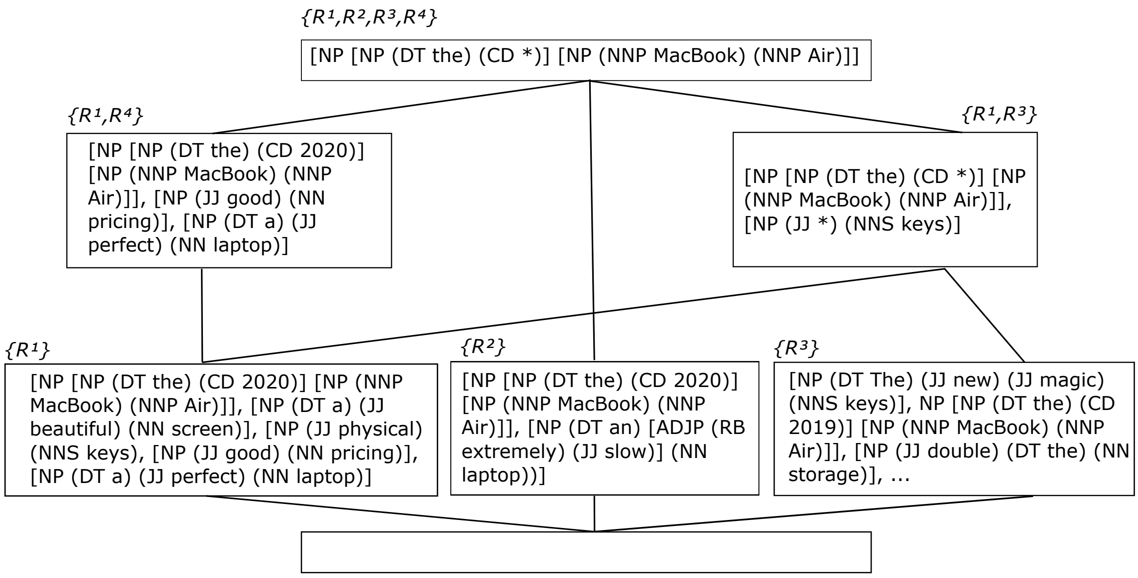


Figure 2: Diagram of the pattern structure lattice for the reviews

At the first level of the lattice diagram, we get concepts that correspond to the reviews themselves. In our running example we can see that the reviews are about some specific laptop and its characteristics: [NP [NP (DT the) (CD 2020)] [NP (NNP MacBook) (NNP Air)]], [NP (DT a) (JJ beautiful) (NN screen)], [NP (JJ physical) (NNS keys)]. At the next level, we observe a pairwise intersection of the reviews. There, the description of the review R_4 includes the description of the review R_1 ($\delta(R_4)$ is more general than ($\delta(R_1)$), and the description of the review R_2 is not supported by any other review. The top-level concept contains phrases, which are common for all texts. In this case, the most common phrase characterizes the brand of the laptop.

As we can see the initial noun phrases revealed for the product (the union of the descriptions from the first level of the lattice) were aggregated into two more general groups. The first group consists of the common description of the reviews R_1 and R_4, and the second group is the common descriptions of R_1 and R_3. We propose to enlarge the initial data about the product with the descriptions that are supported by at least 30% of the reviewers. In our case, these are the phrases from the second and the following levels of the lattice. This aggregation of the phrases allows us to reduce the number of informative characteristics of the product by omitting the ones, which are rarely mentioned in the reviews.

We want to highlight one more detail. In the above example, the intersection operation has been defined as the strict match of the part of speech without consideration of the words themselves. However, we lose lots of important information that is decoded in the words, the sentiment or synonymy, for instance, if we have two noun phrases such as *"fast processor"* and *"quick processor"*, the intersection operation defined above provides the result of intersection as *"* processor"*. In this case, we lose the information that both the reviewers were satisfied with the speed of the processor's work.

We assume that considering the contextual word embedding provided by Bert, e.g., could improve the performance of the IR chatbot and avoid losing the important information during the text features retrieval. Thus, when we calculate the intersection between two noun phrases we should compare not just the part of speech tags, but also the embeddings of the adjectives and the nouns in order to keep the synonymy among the phrases.

4 Combination of the textual and catalog features and building the knowledge model

In the previous sections we described the preparation of text data that can be extracted from the users' reviews (or some other text descriptions of the product). Thus, we enlarge standard characteristics of the product that could be revealed from the database with its text descriptions. Based on this data we propose to build the conceptual knowledge model that will represent the data in the form of non-overlapping hierarchically organised groups of objects and their common descriptions. We have already implemented the similar approach in data aggregation block (Section 3.3.1). To build this knowledge model we also leverage pattern structures, whereas, now the set of objects is the real set of the products available in the dataset, and the description is the row in this dataset containing all the features that the system possesses about the product: the numerical, categorical, binary, and textual ones.

Based on the constructed knowledge model, the chatbot walks through the lattice and navigate the user to the group of the objects that satisfies his initial query. Simultaneously the chatbot assesses the variability of the items inside the closed concept and proposes some attributes for the refinement.

In the framework proposed in (Makhalova et al., 2019) the authors assumed that the user specifies only the numerical characteristics of the product, and the chatbot navigates the user and refines only this type of attributes, such as price, etc. Now, we can take into account both the numerical and text description of the product, and, thus, to expand the user's scope to formulate his query.

5 Navigation procedure

Once the conceptual model is built, the chatbot utilizes this information and traverses the model based on the users' requests. At the beginning of the search, the user formulates the initial query about the product he wants to purchase. This query may contain object and attribute names, values of the attributes, or object description formulated as some phrase. As a result of running the keyword query, we get the set of objects and their descriptions the user has requested, (O_q, d_q), where O_q is a set of objects satisfying this query, and d_q is the description specified by the user. From this pair, we identify the initial lattice node (O_0, d_0) whose description satisfies d_q. If the user mentioned some numerical features, the system matches the numerical values from d_q with the corresponding values in d_0. Otherwise, if the user inserted a phrase, the system retrieves its noun phrases and matches it with the textual description of the products inside a concept (O_0, d_0).

Here we present a model of the interactive search where the chatbot clarifies the user needs in the course of navigation. The chatbot looks for the concept, which satisfies the initial user's request and simultaneously keeps the variability of the attributes that the user did not refine. The specification (updating the set of constraints) for the queries continues until the user found no more appropriate specification or a product that corresponds exactly to what he searched for.

Once (O_0, d_0) is identified, we fix the lattice node and the chatbot shows the user its current position in the lattice (O_i, d_i). The chatbot calculates the diversity of the features inside the concept (Makhalova et al., 2019). If the feature is diverse then the chatbot asks the user, if he wants to refine it. Then at each iteration i the user is expected to request one of the following:

1) *update*

   - specify some attributes proposed by the chatbot for refinement, thus the chatbot will reduce the set of the object of interest $O_i \to O_{i+1}$. That can be requested via the utterance *"attribute name" - "refined value"*, or *"attribute name" is more than/less than "refined value"*.
     This is a jump down the lattice as we have fewer objects and less general description.
   - specify some attributes that the chatbot did not propose, however the user decided to refine it. This is a jump down the lattice as we have fewer objects and less general description.
   - return to the previous step and to decline some of the refinements. For example, the user has refined the color of the product as *"green"*, however, he decided to look at the objects which are *"green or blue"*.
     This is a jump up the lattice as we have more objects and more general description.

2) *abrupt change of interest*

- Proceed to new set O_{i+1} . That can be requested via the utterance 'switch to O_{i+1}
- Run new query and go to (O_q, d_q) All of these three cases is a long-distance jump to a new area in the lattice.

3) *text description*

- The user may formulate its request in natural language. In this case, the system reveals the noun phrases from the parsed request and finds the correspondence among them and textual descriptions of the concepts in the knowledge model.

6 Simulation of the dialogue

In this section, we simulate the interaction of a chatbot with a user, where the knowledge model is built based on the information obtained from the small dataset presented in Table 1 (see Appendix). The textual data was prepared in accordance with the technique described in Section 3.

The knowledge model obtained for this dataset and the detailed description of some concepts is given in the Appendix.

Below we present two dialogues with the chatbot that walks through the constructed knowledge model and helps the user to purchase a tablet based on his requests.

Dialogue 1	Dialogue 2
User I am looking for a fast tablet. Apple.	**User** Tablet with the weight less than 0.5.
Bot *We have found 3 objects satisfying your requirements.* **Tablet Apple iPad Pro 12.9 2018**, **Tablet Apple iPad Mini 2019**, **Tablet Apple iPad Air 10.5 2019** Would you like to specify some of the characteristics: the price ([384.0, 960.0]), weight ([0.3, 0.631]), or screen size ([7.9, 12.9])?	**Bot** *We have found 4 objects satisfying you requirements.* **Tablet Apple iPad Pro 12.9 2018, Tablet Apple iPad Air 10.5 2019, Tablet Apple iPad 10.2, Tablet Samsung Galaxy TAB A 8.0 2019** Would you like to specify some of their characteristics price ([384.0, 960.0]), color ([0.3, 0.631]), brand ([Apple, Samsung]), screen size ([7.9, 12.9])?
User The price should be less than 800.	**User** The screen size should be more than 10, and I also want to utilize this *tablet for work*.
Bot *We have found 2 objects satisfying you requirements.* **Tablet Apple iPad Mini 2019**, **Tablet Apple iPad Air 10.5 2019** Would you like to specify some of the characteristics: the battery ([5124.0; 8134.0]), weight ([0.3, 0.456]), screen size ([7.9, 10.5])?	**Bot** *We have found 1 object satisfying you requirements.* **Tablet Samsung Galaxy TAB A 8.0 2019** Do you want to purchase it, or start a new session?
User No, but it should also have a bright screen.	**User** Is it *a reliable tablet*?

Dialogue 1	Dialogue 2
Bot *Object* **Tablet Apple iPad Air 10.5 2019** *satisfies the requirements*. Do you want to purchase it, or start a new session?	**Bot** Unfortunately, we could not found the phrase "a reliable tablet" among the object descriptions, however, you could look at some reviews in more details.
User Purchase.	**User** No.

As we can see, the query formulated by the user does not contain any attribute names or their values, it is a simple phrase in natural language. So, we reveal the set of noun phrases from the query (*[NP (NNP Apple)], [NP (DT a) (JJ fast) (NNP tablet)]*), and finds the corresponding concept, whose text description contains these phrases. This is the second concept, which contains three tablets (see Appendix). The system calculates the varied characteristics and proposes them to the user for refinement. In the next step, the user has chosen to specify the price. So, the chatbot walks down the lattice to the less general fifth concept and proposes new features for the refinement. The user refuses to specify the proposed features, but refines the new textual characteristics, so the chatbot moves down the lattice and finds the next less general concept, whose textual description satisfies the noun phrase *"a bright screen"*.

During the second dialogue, the user introduces the noun phrase, that is not contained in the description of any object. So, the chatbot tells the user that this phrase is not contained in the text description of the objects in the database, however, he might show the user text descriptions of the current objects, or he may want to specify another characteristic of an object. The user refuses, so, now he can start a new search session.

7 Conclusion

In the work, we have introduced a conceptual-based IR chatbot that is able to process both the standard numerical features of an object and the features retrieved from some textual descriptions.

Our work is motivated by the fact that utilizing a clear navigational structure such as lattice improves the performance of IR systems. Despite the fact that the IR systems utilizing these structures exist, none of them combines the standard numerical features with the complex features of the product that can be extracted from texts. So, in our work, we are concentrated on retrieving text features from the dataset of users' reviews, which are a great source of additional information about an item. The pipeline we introduced combines three steps of text data preparation, where we filter, extract, and aggregate the text descriptions. Thus, the final text characteristics of the object are obtained only from the argumentative texts that provide the reasons and motivation rather than plain sentiment. The features themselves are aggregated into the groups in accordance with the percentage of the reviewers that have used them to describe a product.

The introduction of the textual features to the product description enables the chatbot to process not just the standardized users queries written in the *"Attribute - value"* form, but also the requests formulated in natural language.

We also presented the simulation of the dialogue that illustrates possible scenarios of the user-chatbot interaction. While we remark that this is a preliminary study, and in the future works we will present the comparison of the proposed framework with the existing models of IR chatbots.

References

Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. *CoRR*, abs/1603.02754.

Boris Galitsky, Dmitry Ilvovsky, Fedor Strok, and Sergei Kuznetsov. 2013. Improving text retrieval efficiency with pattern structures on parse thickets. *CEUR Workshop Proceedings*, 977:6–21, 01.

Boris A. Galitsky, Dmitry I. Ilvovsky, and Sergey O. Kuznetsov. 2018. Detecting logical argumentation in text via communicative discourse tree. *Journal of Experimental Theoretical Artificial Intelligence*, 30:637 – 663.

Bernhard Ganter and Sergei O. Kuznetsov. 2001. Pattern structures and their projections. In *Conceptual Structures: Broadening the Base*, pages 129–142, Berlin, Heidelberg. Springer Berlin Heidelberg.

Xiang Gao, Sungjin Lee, Yizhe Zhang, Chris Brockett, Michel Galley, Jianfeng Gao, and William B. Dolan. 2019. Jointly optimizing diversity and relevance in neural response generation. In *NAACL-HLT*.

Freya Hewett, Roshan Rane, Nina Harlacher, and Manfred Stede. 2019. The utility of discourse parsing features for predicting argumentation structure. pages 98–103, 01.

Yangfeng Ji and Jacob Eisenstein. 2014. Representation learning for text-level discourse parsing. volume 1, pages 13–24, 06.

Sergei Kuznetsov. 1993. A fast algorithm for computing all intersections of objects in a finite semi-lattice. *Automatic Documentation and Mathematical Linguistics*, 27:11–21, 01.

Tou Nh H. Lin, Z. and M. Kan. 2014. A pdtb-styled end-to-end discourse parser. volume 20, pages 151–184.

Tatiana P. Makhalova, Dmitry A. Ilvovsky, and Boris A. Galitsky. 2019. Information retrieval chatbots based on conceptual models. In *Graph-Based Representation and Reasoning*, pages 230–238, Cham. Springer International Publishing.

William C. Mann and Sandra A. Thompson. 1988. Rhetorical structure theory: Toward a functional theory of text organization. *Text Talk*, 8:243 – 281.

Julian J. McAuley and Jure Leskovec. 2013. Hidden factors and hidden topics: understanding rating dimensions with review text. In *RecSys '13*.

Elena Musi, Tariq Alhindi, Manfred Stede, Leonard Kriese, Smaranda Muresan, and Andrea Rocci. 2018. A multi-layer annotated corpus of argumentative text: From argument schemes to discourse relations. In *LREC*.

Marco Passon, Marco Lippi, Giuseppe Serra, and Carlo Tasso. 2018. Predicting the usefulness of amazon reviews using off-the-shelf argumentation mining. pages 35–39, 01.

Fedor Strok, Boris Galitsky, Dmitry Ilvovsky, and Sergei Kuznetsov. 2014. Pattern structure projections for learning discourse structures. In Gennady Agre, Pascal Hitzler, Adila A. Krisnadhi, and Sergei O. Kuznetsov, editors, *Artificial Intelligence: Methodology, Systems, and Applications*, pages 254–260, Cham. Springer International Publishing.

Dandison Ukpabi and Heikki Karjaluoto, 2019. *Chatbot Adoption in Tourism Services: A Conceptual Exploration*, pages 105–121. 10.

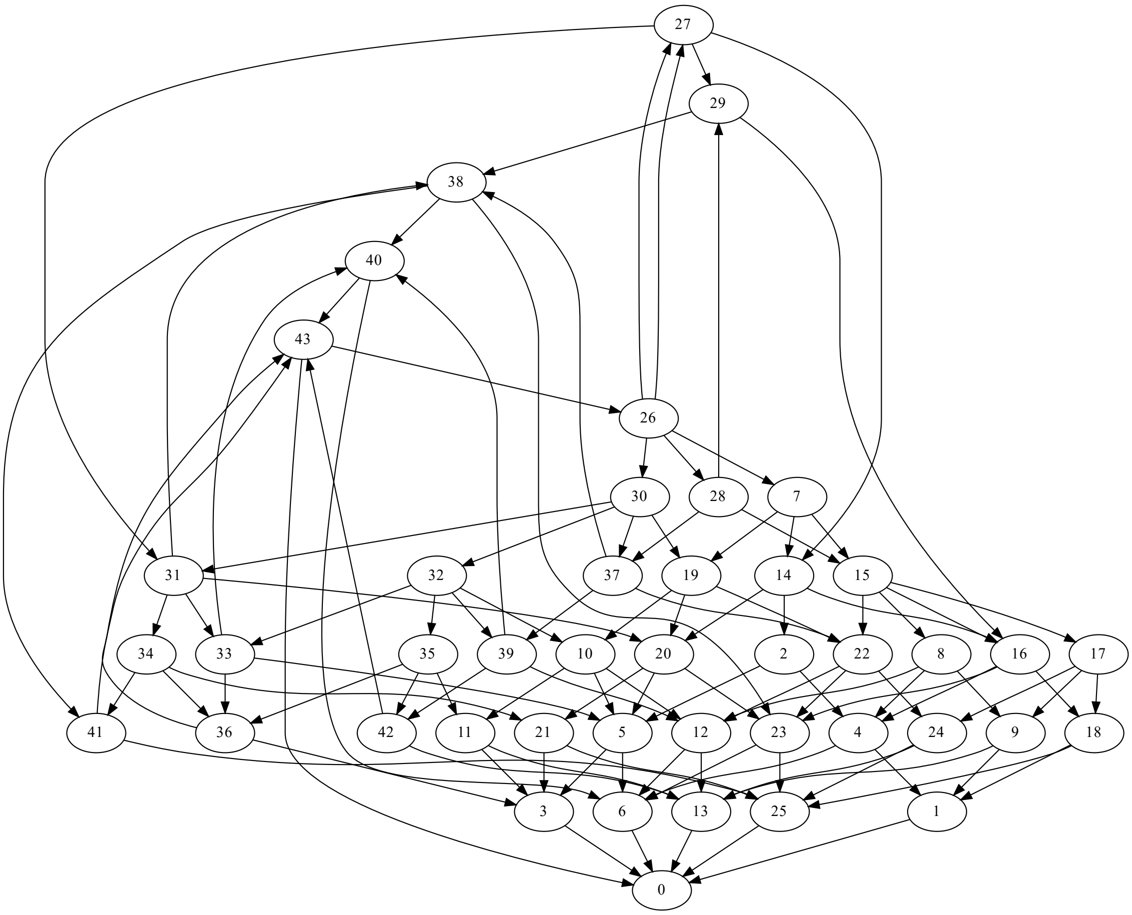
Dean van der Merwe, Sergei Obiedkov, and Derrick Kourie. 2004. Addintent: A new incremental algorithm for constructing concept lattices. In Peter Eklund, editor, *Concept Lattices*, pages 372–385, Berlin, Heidelberg. Springer Berlin Heidelberg.

Appendix

Below there is a fragment of the dataset containing the information about 6 tablets. The features of the tablets are presented as the standard numerical characteristics and their text descriptions. The graph given after the table represents the knowledge model (in the form of the concept lattice). This is a hierarchically organized set of pairs, called concepts. Each pair consists of the set of objects and their common descriptions. The concepts with the lower numbers are more general (they contain fewer objects and their description is less variable) than the ones with the higher numbers (they contain more objects and their description is more variable).

Table 1: A fragment of tablet database

	Brand	Battery	Weight	Back camera	OS	Hard memory	Size	4G LTE	Sensor	Color	Price	Text
Tablet Apple iPad Pro 12.9 2018	8	9720	0.631	12	4	4	12.90	0	1	1, 5, 6, 7, 8	960	(Apple), (fast tablet), (work) ...
Tablet Apple iPad Mini 2019	8	5124	0.3	8	4	3	7.90	0	1	1, 5, 6, 7, 8	384	(Apple), (* tablet) ...
Tablet Apple iPad Air 10.5 2019	8	8134	0.456	8	4	3	10.50	0	1	1, 5, 6, 7, 8	612	(* bright screen), (Apple), (fast tablet), (for work) ...
Tablet Dell Latitude 7200	5	5250	0.851	8	1	16	12.30	1	1	1, 8	2281	(movies and videos), ...
Tablet Apple iPad 10.2	8	8827	0.483	8	6	3	12.20	0	1	1, 5, 6, 7, 8	400	(Apple), (* tablet), , (for work) ...
Tablet Samsung Galaxy TAB A 8.0 2019	7	5100	0.345	8	5	2	8.00	0	1	1, 8	150	(for work), ...



Below are the concepts that the chatbot has passed during the communication with the user in the first and second dialogues respectively.

Dialogue 1	
Concept at position 2	
Feature	Value
Brand	Apple
Battery	[5124.0; 9720.0]
Weight	[0.3; 0.631]
Warranty	1
Back camera	[8, 12]
OS	Mac os
Hard memory	[3, 4]
Screen size	[7.9, 12.9]
4G LTE	No
Sensor	1
Color	[1,5,6,7,8]
Price	[384.0, 960.0]
Text	[NP (JJ *) (NN tablet)], [NP (JJ *) (NNS sensor)], [NP (NNP Apple)]

Dialogue 1	
Concept at position 5	
Feature	Value
Brand	Apple
Battery	[5124.0; 8134.0]
Weight	[0.3; 0.456]
Warranty	1
Back camera	8
OS	Mac os
Hard memory	3
Screen size	[7.9, 10.5]
4G LTE	No
Sensor	1
Color	[1,5,6,7,8]
Price	[384.0; 612.0]
Text	[NP (JJ *) (NN tablet)], [NP (NNP Apple)]

Dialogue 2	
Concept at position 31	
Feature	Value
Brand	Apple, Samsung
Battery	[5100.0; 8827.0]
Weight	[0.3; 0.483]
Warranty	1
Back camera	8
OS	Android, iOS 10, iOS last
Hard memory	[2.0; 3.0]
Screen size	[7.9; 12.2]
4G LTE	No
Sensor	1
Color	[1, 8]
Price	[150.0; 612.0]
Text	[NP (NN device)]

Dialogue 2	
Concept at position 43	
Feature	Value
Brand	Samsung
Battery	[5100.0; 5100.0]
Weight	[0.345; 0.345]
Warranty	1
Back camera	8
OS	Android
Hard memory	2
Screen size	[8.0, 8.0]
4G LTE	No
Sensor	1
Color	[1,8]
Price	[150.0; 150.0]
Text	[NP (NN tablet) (NNP samsung) (NNP galaxy)], [NP (DT a) (NN laptop)) (PP (IN for) (NP (NN work))]