

# Chatbot with a Discourse Structure-Driven Dialogue Management

**Boris Galitsky**  
Knowledge Trail Inc.  
San Jose, CA, USA

bgalitsky@hotmail.com

**Dmitry Ilvovsky**  
National Research University  
Higher School of Economics  
Moscow, Russia

dilvovsky@hse.ru

## Abstract

We build a chat bot with iterative content exploration that leads a user through a personalized knowledge acquisition session. The chat bot is designed as an automated customer support or product recommendation agent assisting a user in learning product features, product usability, suitability, troubleshooting and other related tasks. To control the user navigation through content, we extend the notion of a linguistic discourse tree (DT) towards a set of documents with multiple sections covering a topic. For a given paragraph, a DT is built by DT parsers. We then combine DTs for the paragraphs of documents to form what we call extended DT, which is a basis for interactive content exploration facilitated by the chat bot. To provide cohesive answers, we use a measure of rhetoric agreement between a question and an answer by tree kernel learning of their DTs.

## 1 Introduction

Modern search engines have become very good at understanding typical, most popular user intents, recognizing topic of a question and providing a relevant links. However, search engines are not necessarily capable of providing an answer that would match a style, personal circumstances, knowledge state, an attitude of a user who formulated a query. This is particularly true for long, complex queries, and for a dialogue-based type of interactions. In a chat bot, the flow *query – clarification request – clarification response – candidate answer* should be cohesive, not just maintain a topic of a conversation. Moreover, modern search engines and modern chat bots are unable to leverage an immediate, explicit user feedback on

what is most interesting and relevant to them.

The chat bot we introduce in this demo paper is inspired by the idea that knowledge exploration should be driven by navigating a single discourse tree (DT) built for the whole corpus of relevant content. We refer to such tree as *extended discourse tree*. Moreover, to select rhetorically cohesive answers, chat bot should be capable of classifying question-answer pairs as cohesive or not. This can be achieved by learning the pairs of extended discourse trees for the question-answer pairs. A question can have an arbitrary rhetoric structure as long as the subject of this question is clear to its recipient. An answer on its own can have an arbitrary rhetoric structure. However, these structures, the DT of a question and the DT of an answer should be correlated when this answer is appropriate to this question. We apply a computational measure for how logical, rhetoric structure of a request or a question is in agreement with that of a response, or an answer.

Over last decade, Siri for iPhone and Cortana for Windows Phone have been designed to serve as digital assistants. They analyze input question sentences and return suitable answers for users queries (Crutzen et al., 2011). However, they assume patterned word sequences as input commands. Moreover, there are previous studies that combine natural language processing techniques with ontology technology to implement computer system for intellectual conversation. There are chat bot systems including ALICE3, which utilizes an ontology, like Cyc, API.ai and Amazon Lex, the platforms for developers to build chat bots. Most of these systems expected the correct formulation of a question, certain domain knowledge and a rigid grammatical structure in user query sentences; however, less rigid structured sentences can appear in a users utterance to a chat bot. Developers of chat bot platforms

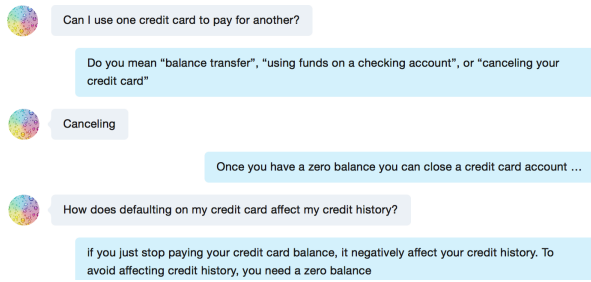


Figure 1: Sample dialogue. User questions and responses are aligned on the left and system responses - on the right.

can specify preset Q/A pairs, but are unable to introduce domain-independent rhetoric constraints which are much more flexible and reusable.

## 2 Personalized and Interactive Domain Exploration Scenarios

Most chat bots are designed to imitate human intellectual activity maintaining a dialogue. The purpose of the chat bot with iterative exploration is to provide most efficient and effective information access for users. A vast majority of chat bots nowadays, especially based on deep learning, try to build a plausible sequence of words (Williams, 2012) to serve as an automated response to user query. Such most plausible responses, sequences of syntactically and semantically compatible words, are not necessarily most informative. Instead, in this demo we focus on a chat bot that helps a user to navigate to the exact, insightful answer as fast as possible.

For example, if a user is formulated her query *Can I use one credit card to pay for another*, the chat bot attempts to recognize a user intent and a background knowledge about this user to establish a proper context. The chat bot leverages an observation learned from the web that an individual would usually want *to pay with one credit card for another to avoid late payment fee when cash is unavailable* as long as the user does not specify her other circumstances.

To select a suitable answer from a search engine, a user first reads snippets one-by-one and then proceeds to the linked document to consult in detail. Reading the answer #n does not usually help to decide which next answer should be consulted, so a user proceeds is to answer #n+1. Since answers are sorted by popularity, for a given user there is no better way than just proceed from

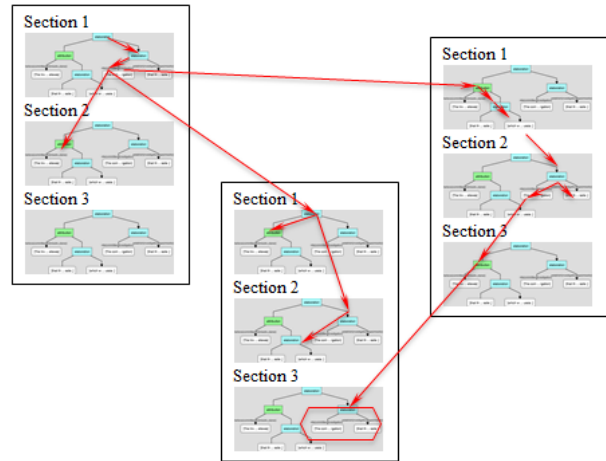


Figure 2: Extended discourse tree for three documents used to navigate to a satisfactory answer

top to bottom on the search results page. On the contrary, the chat bot allows a convergence in this answer navigation session since the answer #n+1 is suggested based on additional clarification submitted after the answer #n is consulted by the user. Chat bot provides *topics* of answers for a user to choose from. These topics give the user a chance to assess how her request was understood on one hand and restore the knowledge area associated with her question on the other hand. In our examples, topics include *balance transfer*, *using funds on a checking account*, or *canceling your credit card*. A user is prompted to select a clarification option, drill into either of these options, or decline all options and request a new set of topics which the chat bot can identify.

## 3 Navigation with the Extended DT

On the web, information is usually represented in web pages and documents, with certain section structure. Answering questions, forming topics of candidate answers and attempting to provide an answer based on user selected topic are the operations which can be represented with the help of a structure that includes the DTs of texts involved. When a certain portion of text is suggested to a user as an answer, this user might want to drill in something more specific, ascend to a more general level of knowledge or make a side move to a topic at the same level. These user intents of navigating from one portion of text to another can be represented in most cases as coordinate or subordinate discourse relations between these portions.

Rhetorical Structure Theory (RST), proposed

by (Mann and Thompson, 1988), became popular as a framework for parsing the discourse relations and discourse structure of a text, represents texts by labeled hierarchical structures such as DTs. A DT expresses the author’s flow of thoughts at the level of a paragraph or multiple paragraphs. But DTs become fairly inaccurate when applied to larger text fragments, or documents. To enable a chat bot with a capability to form and navigate a DT for a corpus of documents with sections and hyperlinks, we introduce the notion of ofextended DT. To avoid inconsistency we would further refer to the original DTs as conventional. To construct conventional discourse trees we used one of the existing discourse parsers (Joty et al., 2013).

The intuition behind navigation is illustrated in Fig.2. Three areas in this chart denote three documents each containing three section headers and section content. For section content, discourse trees are built for the text. A navigation starts with the route node of a section that matches the user query most closely. Then the chat bot attempts to build a set of possible topics, which are the satellites of the root node of the DT. If the user accepts a given topic, the navigation continues along the chosen edge. Otherwise, when no topic covers the user interest, the chat bot backtracks the DT and proceeds to the other section (possibly of another documents) which matched the original user query second best. Finally, the chat bot arrives at the DT node where the user is satisfied (shown by hexagon) or gives up and starts a new exploratory session . Inter-document and inter-section edges for relations such as *elaboration* play similar role in knowledge exploration navigation to internal edges of a conventional discourse tree.

#### 4 Relying on Q/A rhetoric agreement to select the most suitable answers

In conventional search approach, as a baseline, Q/A match is measured in terms of keyword statistics such as TF\*IDF. To improve search relevance, this score is augmented by item popularity, item location or taxonomy-based score (Galitsky et al., 2013; Galitsky et al., 2014). The feature space includes Q/A pairs as elements, and a separation hyper-plane splits this feature space. We combine DT(Q) with DT(A) into a single tree with the root Q/A pair (Fig. 3). We then classify such pairs into correct (with high agreement) and incorrect (with low agreement). Having multiple answer

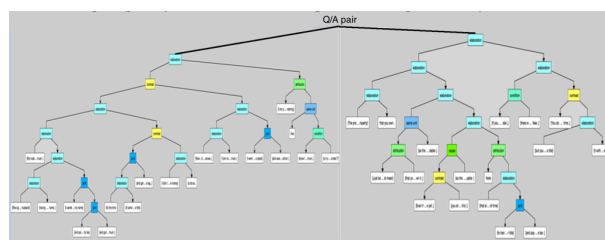


Figure 3: Forming the Request-Response pair as an element of a training set

candidates, we select those with higher classification score. Classification algorithm is based on the tree kernel learning on conventional discourse trees. Their regular nodes are rhetoric relations, and terminal nodes are elementary discourse units (phrases, sentence fragments) which are the subjects of these relations.

We selected Yahoo! Answer set of question-answer pairs with broad range of topics. Out of the set of 4.4 million user questions we selected 20000 which included more than two sentences. Answers for most of the questions are fairly detailed so no filtering was applied to answers. There are multiple answers per questions and the best one is marked. We consider the pair *Question - Best Answer* as an element of the positive training set and *Question - Other Answer* as the one of the negative training set. To derive the negative training set, we either randomly selected an answer to a different but somewhat related question, or formed a query from the question and obtained an answer from web search results.

#### 5 Evaluation

We compare the efficiency of information access using the proposed chat bot with a major web search engines such as Google, for the queries where both systems have relevant answers. For a search engines, misses are search results preceding the one relevant for a given user. For a chat bot, misses are answers which causes a user to chose other options suggested by the chat bot, or request other topics.

Topics of questions include personal finance. Twelve users (authors colleagues) asked the chat bot 15-20 questions each reflecting their financial situations, and stopped when they were either satisfied with an answer or dissatisfied and gave up. The same questions were sent to Google, and evaluators had to click on each search results snippet to get the document or a web page and decide on

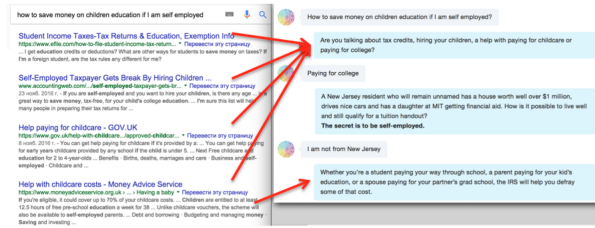


Figure 4: Comparing conventional search engine with chat bot in terms of a number of iterations

whether they can be satisfied with it.

The structure of comparison of search efficiency for the chat bot vs the search engine is shown in Fig. 4. The top portion of arrows shows that all search results (on the left) are used to form the list of topics for clarification. The arrow on the bottom shows that the bottom answer ended up being selected by the chat bot based on two rounds of user feedback and clarifications.

Parameter / search engine	Web search	Chat bot
Av. time to satisfactory search result, sec	45.3	58.1
Av. time of unsatisfactory search session (giving up and starting a new search), sec	65.2	60.5
Av. # of iter. to satisfactory search result	5.2	4.4
Av. # of iter. to unsatisfactory search result	7.2	5.6

Table 1: Comparison of the time spent and a # of iterations for the chat bot and Google search in the domain of personal finance

One can observe (Table 1) that the chat bots time of knowledge exploration session is longer than search engines. Although it might seem to be less beneficial for users, business prefer users to stay longer on their websites, as the chance of user acquisition grows. Spending 7% more time on reading chat bot answers is expected to allow a user to better familiarize himself with a domain, especially when these answers follow the selections of this user. The number of steps of an exploration session for chat bot is 25% lower than for Google search.

## 6 Conclusion

We conclude that using a chat bot with extended discourse tree-driven navigation is an efficient and fruitful way of information access, in comparison with conventional search engines and chat bots fo-

cused on imitation of a human intellectual activity.

The command-line demo<sup>1</sup> and source code<sup>2</sup> are available online under Apache License. Source code is a sub-project of Apache OpenNLP (<https://opennlp.apache.org/>). Since Bing search engine API is actively used for web mining to obtain candidate answers, a user would need to use her own Bing key for commercial use available at <https://datamarket.azure.com/dataset/bing/search>.

## 7 Acknowledgements

This work was supported by Knowledge-Trail Inc., Samur.ai, Sysomos Inc., RFBR grants #16-01-00583 and #16-29-12982 and was prepared within the framework of the Basic Research Program at the National Research University Higher School of Economics (HSE) and supported within the framework of a subsidy by the Russian Academic Excellence Project '5-100'.

## References

- Rik Crutzen, Gjalt-Jorn Y. Peters, Sarah Dias Portugal, Erwin M. Fisser, and Jorne J. Grolleman. 2011. An artificially intelligent chat agent that answers adolescents' questions related to sex, drugs, and alcohol: an exploratory study. *Journal of Adolescent Health*, 48(5):514–519.
- Boris Galitsky, Dmitry I. Ilvovsky, Sergei O. Kuznetsov, and Fedor Strok. 2013. Matching sets of parse trees for answering multi-sentence questions. In *RANLP*, pages 285–293.
- Boris A. Galitsky, Dmitry Ilvovsky, Sergei O. Kuznetsov, and Fedor Strok. 2014. Finding maximal common sub-parse thicketts for multi-sentence search. In *Graph Structures for Knowledge Representation and Reasoning*, pages 39–57. Springer International Publishing.
- Shafiq R. Joty, Giuseppe Carenini, Raymond T. Ng, and Yashar Mehdad. 2013. Combining intra- and multi-sentential rhetorical parsing for document-level discourse analysis. In *ACL (1)*, pages 486–496.
- William C. Mann and Sandra A. Thompson. 1988. Rhetorical structure theory: Toward a functional theory of text organization. *Text-Interdisciplinary Journal for the Study of Discourse*, 8(3):243–281.
- Gwydion M. Williams. 2012. Chatbot detector. *New Scientist*, 215(2873):28.

<sup>1</sup>[https://1drv.ms/u/s!AlZzGY7TCKACHBqHsfVCeNz2V\\_KL](https://1drv.ms/u/s!AlZzGY7TCKACHBqHsfVCeNz2V_KL)

<sup>2</sup><https://github.com/bgalitsky/relevance-based-on-parse-trees>