On a Chat Bot Finding Answers with Optimal Rhetoric Representation

Boris Galitsky[1] **and Dmitry Ilvovsky**[2]
[1]Oracle Inc. Redwood Shores CA USA,
[2]National Research University Higher School of Economics
{boris.galitsky@oracle.com, dilvovsky@hse.ru}

Abstract

We demo a chat bot with the focus on complex, multi-sentence questions that enforce what we call *rhetoric agreement* of answers with questions. Chat bot finds answers which are not only relevant by topic but also match the question by style, argumentation patterns, communication means, experience level and other attributes. The system achieves rhetoric agreement by learning pairs of discourse trees (DTs) for question (Q) and answer (A). We build a library of *best* answer DTs for most types of complex questions. To better recognize a valid rhetoric agreement between Q and A, DTs are extended with the labels for communicative actions. An algorithm for finding the best DT for an A, given a Q, is evaluated.

1. Introduction

Over last few years, chat bots for efficient information access has become fairly popular. However, their relevance is still not as high as desired, which makes conversation longer, less efficient and less informative. For chat bots performing customer support capabilities, low relevance of answers and a lack in overall cohesiveness in support sessions may lead to customer frustration and as a result to a low customer retention.

Modern search engines are good at topical relevance; however, for a chat bot answers should not only be relevant but also need to follow certain style and specific session coherence. A sequence of clarification requests and intermediate answers need to form a logical sequence, not just be constrained by a certain topic or entity. Although an entity-based and topic-based relevance has been extensively explored, coherent flow of utterances in a dialogue in general and a rhetoric correlation between questions is a research area in its infancy.

In this demo paper we improve the cohesiveness of a chat bot session flow by learning correct and incorrect sequences of utterances. When an answer or a clarification expression is weakly rhetorically correlated with a question or a request, a user can identify a dialogue flow broken even when all utterances are about the same entity and therefore topical relevance is high.

The dialogue flow corruption due to a weak rhetoric correlation occurs mostly for complex (Chali et al., 2009; Galitsky, 2017), non-factoid dialogue sessions, involving clarification, explanation, comparison and other means to provide a chat bot user with a complete coverage on a topic or entity. Unsuitability of a request-response sequence usually becomes transparent via the ways an answer is communicated, explained or backed up. Examples of uncorrelated request-response pairs include:

- A chat bot user is a beginner and looks for a general recommendation but an answer is intended for professional, too specific, contains unnecessary arguments and details;
- A user formulated his ambiguous, controversial question relying on a contraposition, but a response just includes a set of plain facts, which are not opinionated;
- A user makes a claim, backs it up and asks for a confirmation. However, an answer just negates it without providing any support.

In the proposed system, rhetoric correlation is achieved via supervised learning of the sequences of utterances. We accumulate a set of correct utterance sequences, including question/answer pairs as our positive dataset, and a set of incorrect sequences and pairs – as the negative one.

Proceedings of Recent Advances in Natural Language Processing, pages 253–259,
Varna, Bulgaria, Sep 4–6 2017.
https://doi.org/10.26615/978-954-452-049-6_035

The proposed chat bot relies on a computational measure for how logical rhetoric structure of a question is correlated with that of an answer. The system forms a discourse tree (DT) representation for a question/answer pair based on Rhetoric Structure Theory (RST, Mann and Thompson, 1988, Figs 1 and 2), learns them and solves a Q/A pair classification problem of relating them into a class of valid (correct, appropriate answer) or invalid pairs.

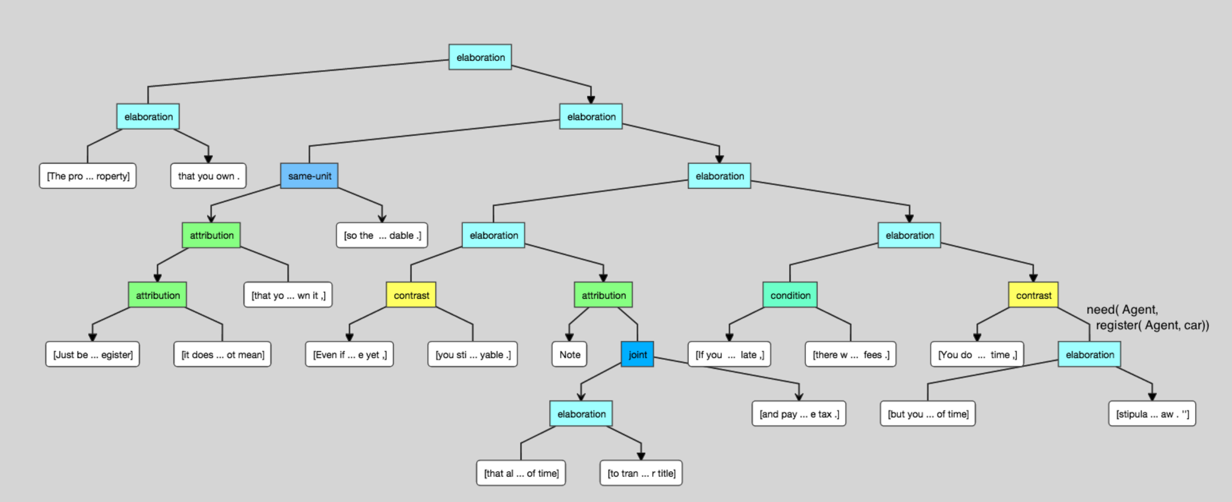


Fig. 1: Full discourse tree for the question

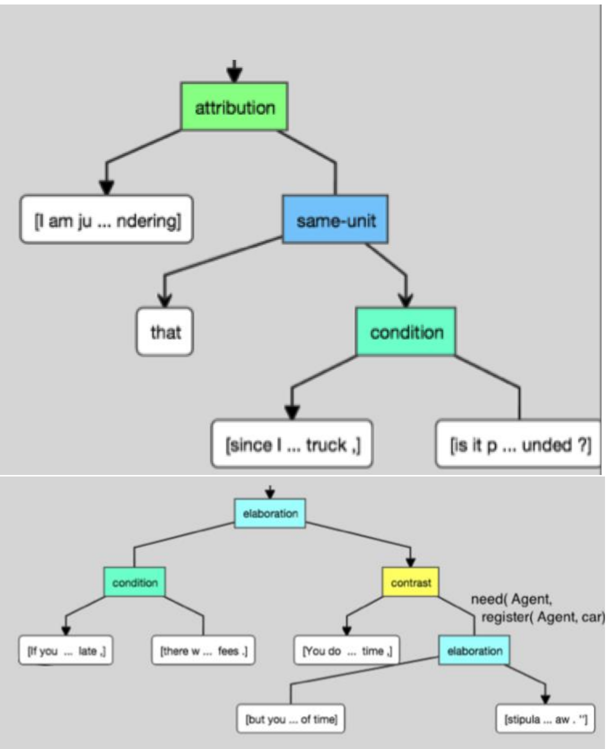


Figure 2: A fragment of a discourse tree for a question (on the top). A fragment of the discourse tree for the A that should be in agreement with this question (on the bottom)

Although the parsers producing discourse trees are constantly improving (Joty et al., 2013; Feng and Hirst, 2014), they identify too few rhetoric relations other than elaboration and joint. The scope of industrial applications of rhetoric-level analysis and rhetoric parsers is fairly restricted to content generation and summarization.

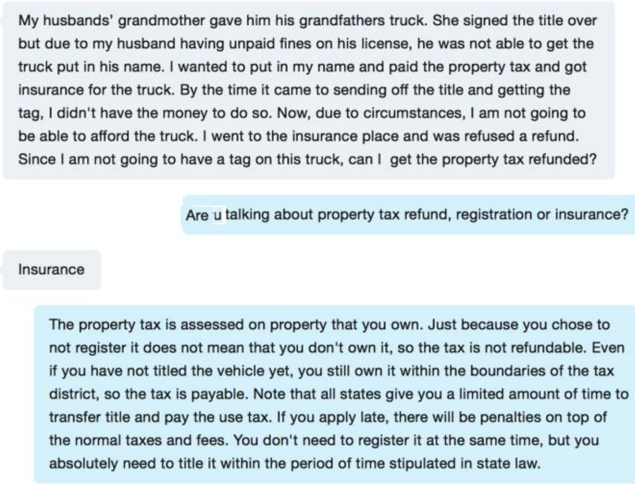
In our previous studies DTs have been found to assist in answer indexing to make search more relevant: query keyword should occur in nucleus rather than a satellite of a rhetoric relation (Galitsky et al., 2015). Hence we developed a chat bot that leverages rhetoric parsing.

A sample chat bot session is available at github.com/bgalitsky/relevance-based-on-parse-trees/blob/master/examples/botSessionExample.txt.

2. Adjusting Rhetoric Representation of Answer to that of a Question

In Fig. 3 we introduce an example of an initial question, a disambiguation request formed by the chat bot, the option selected by a user and the final response. We show that there is a certain correlation between their rhetoric representations, discourse trees for the initial question and the final answers. If a question has a set of certain rhetoric relations, its answer is expected to reflect them in some form. Rhetoric correlation is expected to facilitate a direct, exact and detailed answer. Moreover, rhetoric correlation reduces a number of intermediate disambiguation requests, the length of information access sessions and how many times a user reads irrelevant answers, on the pathway to a satisfactory answer.

The question is about property tax and insurance on a car. This Q is controversial: on one hand, all *owned properties are taxable*, and on the other hand, *the ownership is somewhat incomplete.* This controversy is expected to be addressed by a good answer, relevant not only in terms of topic but also in terms of rhetoric correlation. To provide such A, a responder makes a statement concerning the necessity to pay tax on whatever is owned irrespectively of the registration status.



My husbands' grandmother gave him his grandfathers truck. She signed the title over but due to my husband having unpaid fines on his license, he was not able to get the truck put in his name. I wanted to put in my name and paid the property tax and got insurance for the truck. By the time it came to sending off the title and getting the tag, I didn't have the money to do so. Now, due to circumstances, I am not going to be able to afford the truck. I went to the insurance place and was refused a refund. Since I am not going to have a tag on this truck, can I get the property tax refunded?

Are u talking about property tax refund, registration or insurance?

Insurance

The property tax is assessed on property that you own. Just because you chose to not register it does not mean that you don't own it, so the tax is not refundable. Even if you have not titled the vehicle yet, you still own it within the boundaries of the tax district, so the tax is payable. Note that all states give you a limited amount of time to transfer title and pay the use tax. If you apply late, there will be penalties on top of the normal taxes and fees. You don't need to register it at the same time, but you absolutely need to title it within the period of time stipulated in state law.

Figure 3: Chat bot log with a complex question, its clarification and answer (a demo screen-shot)

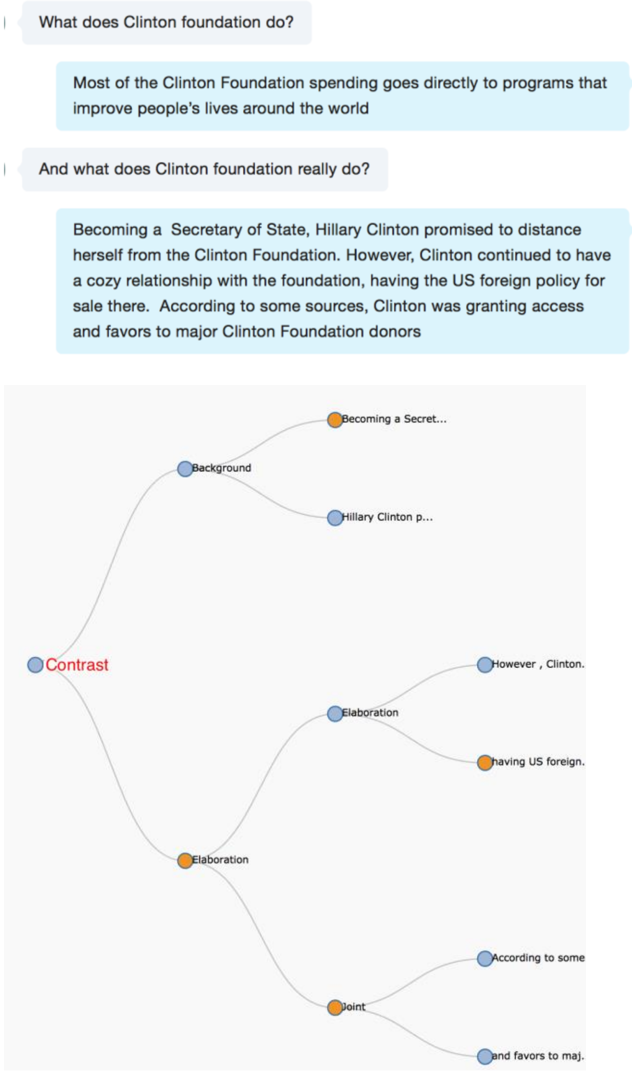
The reader can observe that since the question includes rhetoric relation of *Contrast*, the answer has to match it with a similar relation to be convincing. An adequate answer should clarify the difference between the presence and absence of the *contrast* clauses in the question. Otherwise, this answer would look incomplete. Characteristic sub-trees here are *Contrast-[Elaboration, ..., Elaboration]-Contrast*, which is present in question and therefore should also occur in the answer to match the argumentation style of the question.

A fragment of a DT with a special argumentation pattern used in a question is shown in Fig. 1. When a question is formulated with a chain *Attribution (to an authority)* - *Condition (if my case is covered by certain regulation issued by this authority),* the answer is expected to confirm or reject this *Condition.* Indeed, there is a matching *Condition* relation in the answer, associated with *Elaboration* (Fig. 2, on the left).

To represent the rhetoric agreement better, we augment DT with the information how each discourse unit is communicated. Some edges of communicative DTs are labelled with the expressions for communicative actions (CAs) in the VerbNet (Kipper et. al., 2008) form (shown on the right of Fig. 2). Such DTs are called Communicative Discourse Trees (CDTs).

A high-level view of the full DTs for these Q and A is shown in Fig. 4. Both trees are merged into a single one for the purpose of automated learning of rhetoric agreement: a given pair can be (structurally) similar to Q/A pair of either positive or negative training set. Mapping of two occurrences of *Contrast* relations in Q into that of in A is shown by bolded arcs.

3. Maintaining a Sequence of Discourse trees



What does Clinton foundation do?

Most of the Clinton Foundation spending goes directly to programs that improve people's lives around the world

And what does Clinton foundation really do?

Becoming a Secretary of State, Hillary Clinton promised to distance herself from the Clinton Foundation. However, Clinton continued to have a cozy relationship with the foundation, having the US foreign policy for sale there. According to some sources, Clinton was granting access and favors to major Clinton Foundation donors

Figure 4: A dialogue and a discourse tree for the final answer.

In many cases, a current utterance needs to match the rhetorics of just a previous utterance. However, in many cases not just a previous question but the whole chain needs to be taken into account (Fig.4).

The first utterance is perceived as a factoid (general) question and a common answer is given. However, this answer is followed by a sarcastic form of dissatisfaction, so the same question is repeated with *really* note. Now, instead of providing a general, common answer of the fits-for-all kind, the chat bot finds a controversial opinion about an entity in question. The controversy is expressed via the Contrast relation between the parts of this utterance: On one hand, a person‘s role is a Secretary of State, but on the other hand this person gives special favors to certain people.

A scenario template that is expected to be earned in this case is as follows:

1) A factoid question
2) A general factoid, definitional answer
3) Dissatisfaction with (2), expressed via sarcastic *really*. Repeated question.
4) Opinionated answer to the same question (2).

4. Identifying Rhetoric Correlation

A search engine dealing with complex questions such as Yahoo! Answers needs a systematic approach to assess the rhetoric agreement and select the most suitable answers among the relevant ones. DTs’ features could be represented in a numerical space where a classification into valid or invalid Q/A pairs would be conducted, however structural information on DTs would not be leveraged. Conversely, rhetoric agreement could be measured in terms of maximal common sub-DTs, but it is computationally intensive and too sensitive to errors in DT construction. Therefore a DT-kernel learning approach is selected which applies SVM learning to a set of all sub-DTs of the DT for Q/A pair.

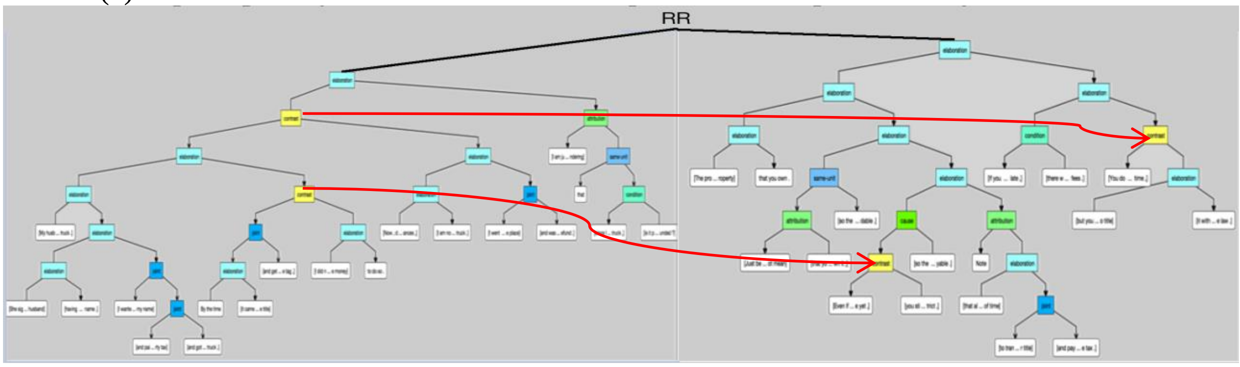


Figure 5: A high level view of the DT-Q – DT-A pair (full discourse trees for the example in Figure 3).

Tree kernel family of approaches is not very sensitive to errors in parsing (syntactic and rhetoric) because erroneous sub-trees are mostly random and will unlikely be common among different elements of a training set.

Given a positive dataset of valid Q/A pairs and a negative dataset of relevant Q/A pairs which are in a poor agreement, we attempt to recognize if a given Q/A pair is valid. Notice that a DT for Q and a DT for A can be arbitrary, but only DTs for a pair can be *valid* or *invalid*. Then this algorithm is applied to perform passage re-ranking of answers to achieve the highest possible rhetoric agreement maintaining relevance.

Tree Kernel (TK) learning for strings, parse trees and parse thickets is a well-established research area nowadays. The parse tree kernel counts the number of common sub-trees as the discourse similarity measure between two DTs. TK relies on the operation of generalization ‘^’ which is applied at the level of parse and discourse trees, phrases, and words. A version of TK has been defined for DT by (Joty and Moschitti, 2014). In (Wang et al., 2013) special form of TK was used for discourse relation recognition. In this study we extend the TK definition for the CDT, augmenting DT kernel by the information on CAs. In Fig. 4 we have DT-A and DT-Q joined at the root, so that counting common sub-trees with nodes from DT-Q and nodes from DT-A provides a quantitative insight on how these DTs are coordinated.

A CDT can be represented by a vector V of integer counts of each sub-tree type (without taking into account its ancestors).

Only RST arcs of the same type of relation (*presentation* relation, such as *Antithesis, subject matter* relation, such as *Condition, and multinuclear* relation, such as *List*) can be matched when computing common sub-trees. We use *N* for a nucleus or situations presented by this nucleus, and *S* for satellite or situations presented by this satellite. *Situations* are propositions, completed actions or actions in progress, and communicative actions and states (including *beliefs, desires, approve, explain, reconcile* and others). Hence we have the following expression for RST-based generalization '^' for two texts $text_1$ and $text_2$: $text_1$ ^ $text_2$ = $\cup_{i,j}$ ($rstRelation_{1i,}$ (…,…) ^ $rstRelation_{2j}$ (…,…)), where $I \in$ (RST relations in $text_1$), $j \in$ (*RST relations in* $text_2$). Further, for a pair of RST relations their generalization looks as follows: $rstRelation_1(N_1, S_1)$ ^ $rstRelation_2$ (N_2, S_2) = ($rstRelation_1$^ $rstRelation_2$)(N_1^N_2, S_1^S_2).

The texts in N_1, S_1 are subject to generalization as phrases. The rules for rst_1^ rst_2 are as follows. If *relation_type*(rst_1) ! = *relation_type*(rst_2) then similarity is empty. Otherwise, we generalize the signatures of rhetoric relations as sentences: *sentence*(N_1, S_1) ^ *sentence* (N_2, S_2) (Iruskieta et al., 2015).

We define CA as a predicate of the form *verb(agent, subject, cause)*, where *verb* characterizes some type of interaction between involved *agents* (e.g., *explain, confirm, remind, disagree, deny*, etc.), *subject* refers to the information transmitted or object described, and *cause* refers to the motivation or explanation for the subject. To handle meaning of words expressing the subjects of CAs, we apply *word2vec* models (Mikolov et al., 2015). When verbs are identified in Qs and As, CAs are formed from the VerbNet patterns (example is shown on the right of Fig.2).

To compute similarity between the subjects of CAs, we use the following rule: If *subject1=subject2*, then *subject1^subject2* = *<subject1, POS(subject1), 1>*.

Otherwise, if they have the same part-of-speech, *subject1^subject2* = *<*, POS(subject1), word2vecDist(subject1^subject2)>*. '*' is a placeholder for word lemma. If part-of-speech is different, generalization is an empty tuple. It cannot be further generalized.

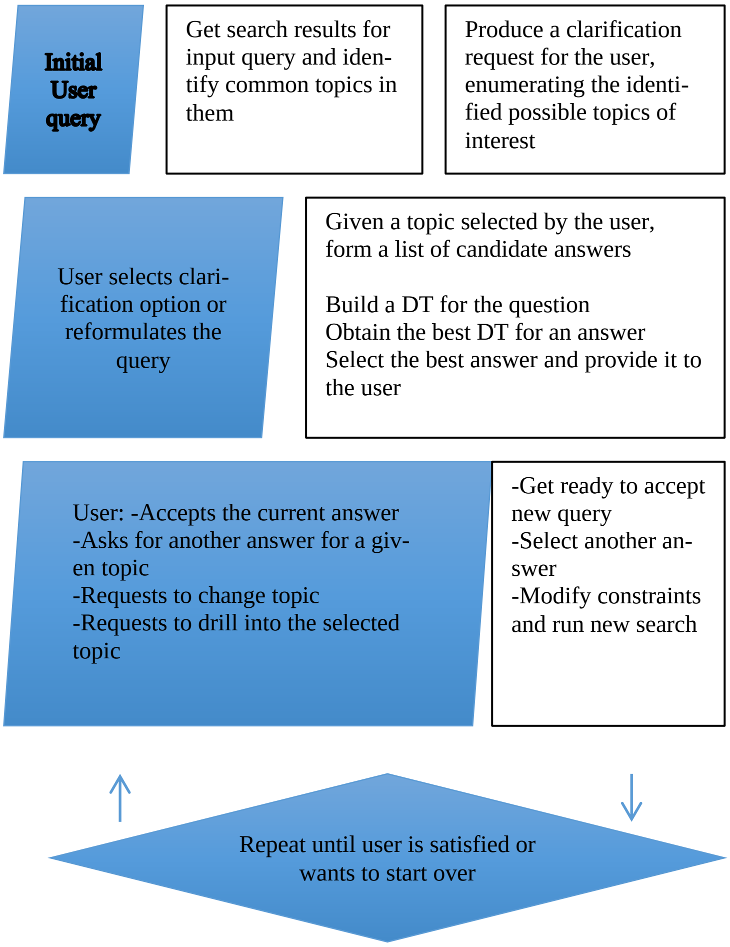
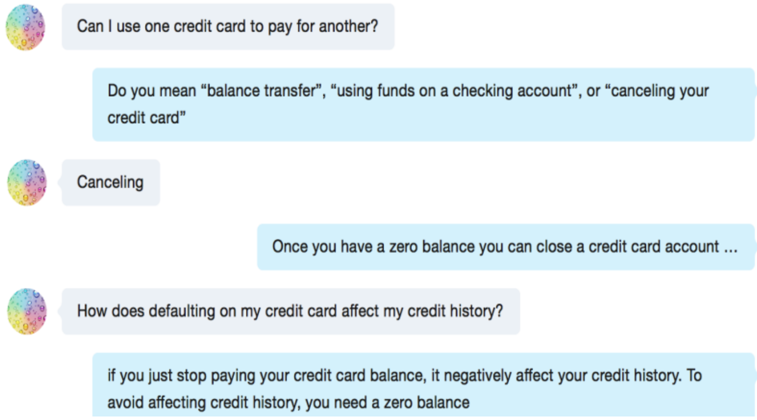

Initial User query
Get search results for input query and identify common topics in them
Produce a clarification request for the user, enumerating the identified possible topics of interest
User selects clarification option or reformulates the query
Given a topic selected by the user, form a list of candidate answers
Build a DT for the question
Obtain the best DT for an answer
Select the best answer and provide it to the user
User: -Accepts the current answer
-Asks for another answer for a given topic
-Requests to change topic
-Requests to drill into the selected topic
-Get ready to accept new query
-Select another answer
-Modify constraints and run new search
Repeat until user is satisfied or wants to start over

Figure 6: System architecture for a chat bot



Can I use one credit card to pay for another?

Do you mean "balance transfer", "using funds on a checking account", or "canceling your credit card"

Canceling

Once you have a zero balance you can close a credit card account ...

How does defaulting on my credit card affect my credit history?

if you just stop paying your credit card balance, it negatively affect your credit history. To avoid affecting credit history, you need a zero balance

Figure 7: Sample dialogue. User questions and responses are aligned on the left and system responses - on the right

5. Chat Bot Implementation

The system architecture for a chat bot which relies on correlation of rhetoric structures of requests and response is shown in Figure 6.

Rhetoric agreement component combines Stanford NLP parsing, coreferences, entity extraction, DT construction (Surdeanu et al., 2013; Joty et al., 2016), VerbNet (Kipper et al., 2008) and Tree Kernel builder (Severyn et al., 2012) are integrated into one system stored at https://github.com/bgalitsky/relevance-based-on-parse-trees. This GitHub page contains the latest links, instructions on how to install resources and run the app.

The chat bot is optimized to minimize the amount of text the user needs to read in the course of getting to the answer. Instead of reading search results snippets and selecting the ones believed to be relevant, the system suggests the topics related to user question and options to drill into them or choose another one. This approach is better suited to personalization then conventional search engines, since user choice such as “financial product for a retired government employee” or “item for a toddler girl” is easier to formalize. A chat bot user is expected to read less and interact & clarify more to have a more efficient information access, which is also appropriate for mobile devices where text tends to be shorter and user interaction faster (Fig. 7).

6. Related Systems and Conclusions

Conventional search engines, from vertical to horizontal, do not use discourse level analysis, as far as the authors know. This is due to its computational load and hard compatibility with big data technologies. Most search engineers consider discourse analysis too abstract and too distant from applications.

Since rhetoric parsers for English has become more available and accurate, their application in search engine indexing is becoming more feasible. As precision and recall of search systems ignoring discourse level information deteriorates, users do not find products, services and information they need, leveraging of linguistic technologies including discourse become realistic for industrial systems.

Most chat bot vendors these days such as Microsoft’s QnA Maker, IBM’s Watson QnA Pair Ranker, botframework.com and Google’s Api.ai provide an NLP chat bot platform. This platform is designed in a way so that the developers from the companies that acquired this platform can feed them with a random set of Q/A pairs and expect a reasonable relevance. Evaluation results for instances of these chat bot platforms are unavailable to the authors, but there is some evidence that the performance of above chat bots is limited when a user deviates from pre-set Q/A pairs.

Another family of chat bots is focused on simulation of intelligent human dialogues on an arbitrary topic instead of sharing searched information or performing a transaction. A number of chat bots are based on deep learning (Cho et al., 2014) of an extended training set of human dialogs. Deep learning-based chat bots are sometimes good at supporting a conversation on an arbitrary topic and building plausible utterances. However, these chat bots are not very suitable for domains such as product selling, financial transaction and customer support due to their non-deterministic nature.

We conclude that selection of answers without taking discourse correlation into account leads to a limited satisfaction with chat bots and search engines. Frequently, answers to user questions obtained from noisy user-generated content are of lower quality. This is true even for a popular content, since search engine optimization algorithms select from an available corpus and optimize the selection for efficient advertisement.

Although most search engines are good at answering popular, a few keyword-size queries, they are not good at answering complex, detailed, multi-sentence questions. The system employs the constructed library of rhetoric structures for answers that match most popular types of long questions. Using this library, a generic chat bot would be capable of providing not only topically relevant but exact, direct, properly backed up, rhetorically appropriate answers.

A web version of the chat bot is available at http://ec2-35-160-183-122.us-west-2.compute.amazonaws.com:8080/Chatbot/indexnew.html.

The bot is implemented in Oracle Mobile Cloud environment https://cloud.oracle.com/mobile.

Acknowledgements

The work of D. Ilvovsky was supported by the Russian Science Foundation under grant 17-11-01294 and performed at National Research University Higher School of Economics, Russia.

References

Chali, Y., Shafiq R. Joty, and Sadid A. Hasan. 2009. *Complex question answering: unsupervised learn-*

ing approaches and experiments. J. Artif. Int. Res. 35, 1 (May 2009), 1-47.

Vanessa Wei Feng and Graeme Hirst. 2014. A linear-time bottom-up discourse parser with constraints and post-editing. In *Proceedings of The 52nd ACL, Baltimore, USA, June*.

Galitsky, B. 2017. Matching parse thickets for open domain question answering, *Data & Knowledge Engineering*, V. 107, pp 24-50.

Galitsky, B. 2013. Machine learning of syntactic parse trees for search and classification of text. *Engineering Application of AI*, 26(3) 1072-91.

Galitsky, B, Ilvovsky, D. and Kuznetsov SO. 2015. *Rhetoric Map of an Answer to Compound Queries*. ACL-2, 681.

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.

Shafiq R. Joty and A. Moschitti. 2014. *Discriminative Reranking of Discourse Parses Using Tree Kernels*. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), page

P. Jansen, M. Surdeanu and P. Clark. 2014. *Discourse Complements Lexical Semantics for Nonfactoid Answer Reranking*. ACL.

Mikel Iruskieta, Iria da Cunha and Maite Taboada. 2015. *A qualitative comparison method for rhetorical structures: identifying different discourse structures in multilingual corpora*. Language Resources and Evaluation, V. 49, I 2, pp 263–309.

William Mann and Sandra Thompson. 1988. Rhetorical structure theory: Towards a functional theory of text organization. Text-Interdisciplinary Journal for the Study of Discourse, 8(3):243–281.

Mikolov, Tomas, Chen, Kai, Corrado, G.S., Dean, Jeffrey. 2015. *Computing numeric representations of words in a high-dimensional space*. US Patent 9,037,464, Google, Inc.

Kipper, K. Korhonen, A., Ryant, N. and Palmer, M. 2008. *A large-scale classification of English verbs*. Language Resources and Evaluation Journal, 42, pp. 21-40.

Mihai Surdeanu, Thomas Hicks and Marco A. Valenzuela-Escarcega. 2015. *Two Practical Rhetorical Structure Theory Parsers*. Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics - Human Language Technologies: Software Demonstrations (NAACL HLT).

Webscope 2017. Yahoo! Answers Dataset. webscope.sandbox.yahoo.com/catalog.php?datatype=l.

Wang, W., Su, J., Tan, C.L. 2010. *Kernel Based Discourse Relation Recognition with Temporal Ordering Information*. ACL.

K. Cho, B. van Merrienboer, C. Gulcehre, F. Bougares, H. Schwenk, Y. Bengio. 2014. *Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation*. EMNLP 2014.

A. Severyn, A. Moschitti. 2012. Fast Support Vector Machines for Convolution Tree Kernels. Data Mining Knowledge Discovery 25: 325-357.