Building Dialogue Structure from Discourse Tree of a Question

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

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

Abstract

In this section we propose a reasoning-based approach to a dialogue management for a customer support chat bot. To build a dialogue scenario, we analyze the discourse tree (DT) of an initial query of a customer support dialogue that is frequently complex and multi-sentence. We then enforce rhetorical agreement between DT of the initial query and that of the answers, requests and responses. The chat bot finds answers, which are not only relevant by topic but also suitable for a given step of a conversation and match the question by style, communication means, experience level and other domain-independent attributes. We evaluate a performance of proposed algorithm in car repair domain and observe a 5 to 10% improvement for single and three-step dialogues respectively, in comparison with baseline approaches to dialogue management.

1 Introduction

Answering questions, a chat bot needs to reason to properly select answers from candidates. In industrial applications of search, reasoning is often substituted by learning from conversational logs or user choices. It helps to make search more relevant as long as a similar question has been asked many times. If there is no data on previous similar question, which is frequently the case, a chat bot needs to apply some form of reasoning to select from candidate answers (Wilks, 1999).

Most frequent type of reasoning is associated with topical relevance. It requires ontology and is domain-specific. Difficulties in building domain ontologies are well known, and in this paper we take a different reasoning-based approach. Once a set of candidate answers or replies is available, how to select most suitable ones? The suitability criteria are two-dimensional: 1) topical relevance; and 2) an appropriateness not associated with topic but instead connected with communicative discourse. Whereas topical relevance has been thoroughly investigated, chat bot's capability to maintain the cohesive flow, style and merits of conversation is an underexplored area.

When a question (Q) is detailed and includes multiple sentences, there are certain expectations concerning the style of an answer (A). Although topical agreement between questions and answers has been extensively addressed, a correspondence in style and suitability for the given step of a dialogue between questions and answers has not been thoroughly explored. In this study we focus on assessment of the cohesiveness of the Q/A flow, which is important for a chat bots supporting longer conversation. When an answer is in a style disagreement with a question, a user can find this answer inappropriate even when a topical relevance is high. Matching rhetorical structures of questions and answers is a systematic way to implement high-level reasoning for dialogue management, to be explored in this work.

A problem in communicative discourse occurs mostly for complex questions (Chali et al., 2009; Galitsky, 2017), arising in miscommunication, a lack of understanding, and requiring clarification, argumentation and other means to bring the answer's author point across. Rhetorical disagreement is associated with a broken dialogue and is usually evident via the means an answer is communicated, explained or backed up.

Our paper is organized as follows. In Section 2 we discuss basic notions of discourse tree text representation. In Section 3 we consider details our approach to building a dialogue based on discourse trees. In Section 4 we present evaluation results for the one of the Q/A tasks.
The system described in this paper is available on our GitHub[1].

[1] https://github.com/bgalitsky/relevance-based-on-parse-trees

Proceedings of the 2018 EMNLP Workshop SCAI: The 2nd Int'l Workshop on Search-Oriented Conversational AI, pages 17–23
Brussels, Belgium, October 31, 2018.
ISBN 978-1-948087-75-9

2 Discourse Tree and Rhetorical Structure

To represent the linguistic features of a text, we used *Rhetorical relations* (RR) between the parts of the sentences, obtained as a *discourse tree*. We relied on Rhetorical Structure Theory (RST, Mann and Thompson, 1988) and deployed state-of-the-art rhetorical parsers (Joty et al., 2013; Surdeanu et al., 2015) to build these discourse trees automatically.

Rhetorical Structure Theory models the logical organization of text, a structure employed by a writer relying on relations between parts of text. RST simulates text coherence by forming a hierarchical connected structure of texts via discourse trees. Rhetorical relations are split into coordinate and subordinate classes; these relations hold across two or more text spans and therefore implement coherence. These text spans are called *elementary discourse units* (EDUs).

Clauses in a sentence and sentences in a text are logically connected by the author. The meaning of a given sentence is related to that of the previous and following sentences. This logical relation between clauses is called the coherence structure of the text. RST is one of the most popular theories of discourse and is based on tree-like discourse structures called discourse trees. The leaves of a DT correspond to EDUs, the contiguous atomic text spans. Adjacent EDUs are connected by coherence rhetorical relations (e.g., *Attribution, Sequence*), forming higher-level discourse units. These units are then also subject to this relation-linking. EDUs linked by a relation are then differentiated based on their relative importance: *nuclei* represent the core parts of the relation, whereas *satellites* represent the peripheral ones.

Let's consider small example of a discourse tree for the text. For the question
"*What does Clinton foundation really do"* one can find the following answer:
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.

The discourse tree of an answer is presented on Figure 1 is based on *Elaboration* and *Background* rhetorical relations.

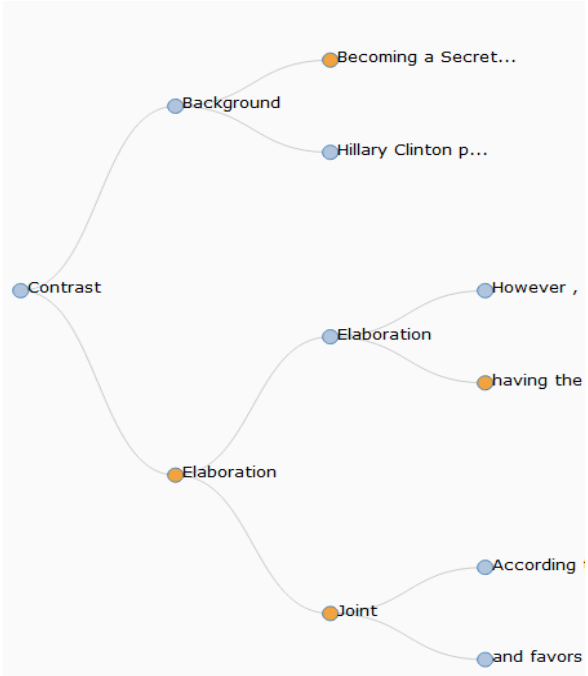

Figure 1: Example of a discourse tree

3 Building Dialogue Structure with Discourse

3.1 Maintaining Discourse in a Dialogue

Once we have a detailed initial question, we frequently can determine which direction we can take a given dialogue. If an answer is formulated in a straight-forward way, then a definitional or factual answer is to follow.

Otherwise, if a question includes a doubt, a request to dig deeper into a topic, or to address a controversy, the dialogue should be handled with replies including attribution, communicating a contrast, explicit handling of what was expected and what actually happened. Hence from Rhetorical relations in initial query the chat bot can select one set of answers over the other not only to cover the main topic, but to also address associated issues raised by the user. It can be done even if the initial query is short and its DT is trivial.

Now imagine for each of answers we obtain multiple candidates, with distinct entities. How the chat bot would know which entity in an answer would be of a higher interest to a user? The chat bot need to include a clarification procedure.

For a single Q/A pair, one can refer to their coordination as rhetorical agreement (Galitsky,

2017). For the dialogue management problem, where a sequence of answers A_i need to be in agreement with an initial question Q, we refer the proposed solution as *maintaining communicative discourse in a dialogue*. It includes three components:

1) Finding a sequence of answers A_i to be in agreement with an initial question Q
2) Maintaining clarification procedure where for each i we have multiple candidate answers and need to rely on a user to select which one to deliver.
3) Allowing the chat bot user to specify additional constraints, formulate more specific questions as answers A_i are being delivered.

3.2 Building Dialogue Structure in Customer Support Dialogues

Let us start with an example of a *customer support dialogue*, where a customer support agent tries to figure out a root cause of a problem (Fig.2.). Customer support scenarios form a special class of dialogues where customers attempt to resolve certain problems, get their questions answered and get to their desired outcomes unreachable using default business procedures. Customer support dialogues frequently start with initial question, a multi-sentence statement of problems Q, from which experienced customer support personal frequently plan a resolution strategy.

The personnel come up with a sequence of recommendations and explanations for them addressing customer concerns expressed in Q. Also, the personnel comes up with some questions to the customer to adjust their recommendations to the needs expressed by the customer in Q. Frequently, due to diverse nature of most businesses, it is hard to find a dialogue in a customer support problem which addresses this exact problem. Therefore, individual answers and recommendations from the previous customer support sessions are used, not the whole such sessions, in the majority of cases. Hence the customer support dialogue management cannot be reduced to the problem of finding sufficiently similar dialogue and just following it: instead, actual construction of a dialogue to address Q is required most of times.

The system finds candidate answers with the keywords and phrases from the initial query, such as *Google Earth, cannot see, attention* and others. Which candidate answers would be the best to match the communicative discourse of the query?

A customer support dialogue can be represented as a sequence:

$$Q, A_1, C_1, A_2, C_2, \ldots,$$

where Q is an initial query describing a problem, A_1 is an initial recommendation and also a clarification request, C_1 is a response to this request, A_2 is a consecutive recommendation and clarification request, C_2 is a response to A_2 and possibly a further question, and so forth. Our goal is to simulate a broad spectrum of dialogue structures via correspondence of discourse trees of utterances. This way once Q is given, the chat bot can maintain the sequence of answers A_i for Q.

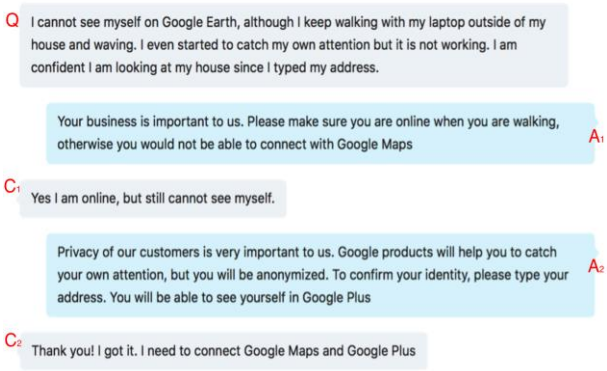

Figure 2: An example of a customer support dialogue

3.3 Finding a Sequence of Answers to be in Agreement with Question

DT for the Q, and DT for the sequence of two answers A_1 and A_2 from our example are shown in Fig. 3. Arrows show which chains of DT-Q determine which chains of DT-A_i.

We will now demonstrate that a *chain* of nodes in DT-Q is determining a corresponding chain of nodes in DT-A. This chain is defined as a path in a DT. The chain of RRs with entities are *Elaboration [see myself Google Earth]-Contrast [walk laptop house]-Temporal [waiving]* on the top of DT-Q is addressed by the chain *Elaboration [online]-Same_Unit [walking]-Contract [Otherwise, not able connect*] in the first answer A_1. We use the label *RR [abbreviated phrase]* for each node of a chain in DT. Notice that not only *RRs* are supposed to be coordinated but the entities in *phrases* as well.

The second answer A_2 attempts to address in a complete way the issues raised in the second part of Q. The first mapping is between the chain RR *Elaboration [catch my attention] -Contrast [not working]* in Q and the chain *Elaboration [catch my attention] - Contrast [anonymized]*.

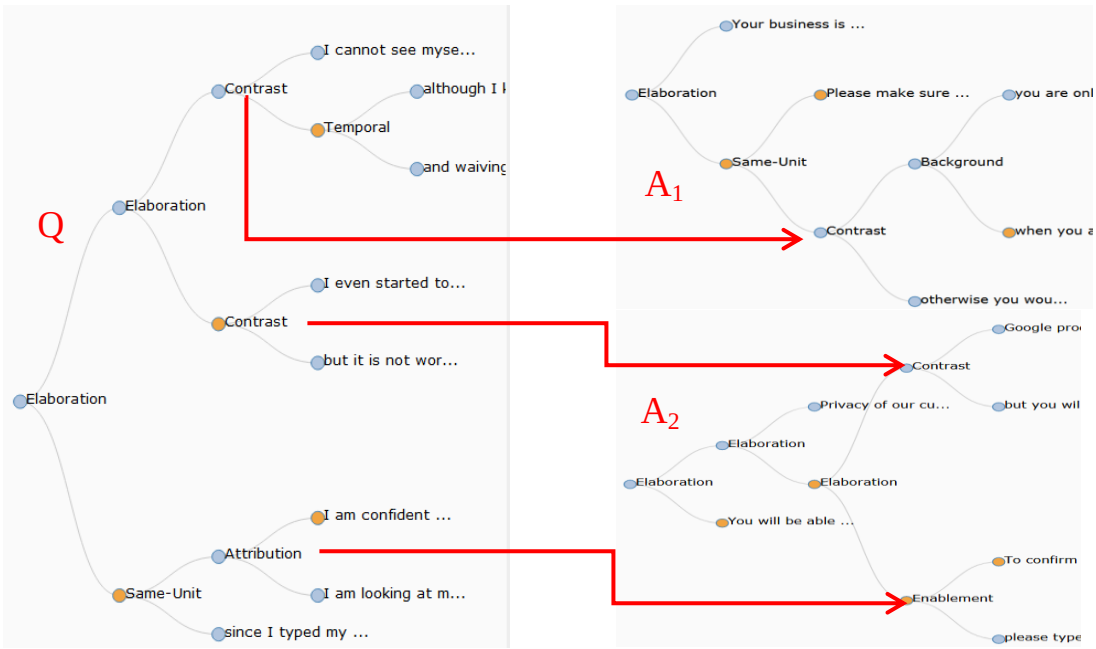

Figure 3: Discourse tree of a question *Q* (on the left) and a sequence (pair) of combined discourse trees (on the right) for the answers A_i.

The main observation here is that the question itself gives us a hint on a possible sequence of answers, or on the order the issues in the question are raised. One can look at the *DT-Q* and form a dialogue scenario (*first do this, obtain confirmation, then do that* ...). Since a dialogue is built from available answer fragments (e.g. from conversational logs), we take candidate answers, form candidate DTs from them and see if they match DT-Q. Hence a single nontrivial DT-Q determines *both* $DT\text{-}A_1$ and $DT\text{-}A_2$. We refer to this capability as *determining the structure of a dialogue* (the structure of a sequence of answers) by the initial Q. We intentionally selected this anecdotal, meaningless example of a customer support dialogue to demonstrate that a full "understanding" of a query is not required; instead, the logical structure of inter-relations between the entities in this query is essential to find a sequence of answers.

Is it possible to come up with a rule for DT-A_i given DT-Q, to formalize the notion of "addressing" an issue in *Q* by an *A*? A simple rule would be for a chain of rhetorical relations for an *A* to be a sub-chain of that of a *Q*, also maintaining respective entities. But this rule turns out to be too restrictive and even invalid in some cases. Our observation is that *DT–A* does not have to copy *DT-Q* or its parts, but instead have some complementarity features. There are two types of considerations for $DT\text{-}A_i$:

1) Each nontrivial RR in Q needs to be addressed by a RR in $DT\text{-}A_i$.
2) There should be a rhetorical agreement between *Q* and A_i, defined for a search engine.

Whereas rhetorical agreement introduces a pair-wise constraint that can be learned from examples of good and bad *Q*/*A* pairs (Galitsky, 2017), we extend it to one-to-many relation between a single Q and a sequence of A_i.

For an RR in $DT\text{-}A_i$ to address an RR in Q, it does not necessarily need to be the same RR but it should not be a default RR such as *Elaboration* or *Joint. Attribution* and *Enablement*, for example, can address *Contrast.*

Also, for a $RR(EDU_{q1}, EDU_{q2})$ in *Q* to be covered by $RR(EDU_{ai1}, EDU_{ai2})$ in A_i, entities *E* should be shared between EDU_{q1} and EDU_{ai1} : $EDU_{q1} \cap EDU_{ai1} = E : E \neq \varnothing$.

3.4 Searching for the Answers for Dialogue construction

Once we established the rules for addressing RRs in *Q*, we can implement search for a series of answers A_i given *Q*. Assuming we have a corpus of dialogues with utterances tagged as *A* or *Q*, it should be indexed offline in at least two follow-

ing fields: 1) keywords of *A* and 2) RRs with their EDUs.

Then once we receive *Q*, build *DT-Q*, and split *DT-Q* into subtrees each of which contains at least single non-default RR. Then for each *subtree-DT-Q* we form a query against these fields:

1) Keywords from the *EDU-subtree-DT-Q;*
2) Non-default RR from *subtree-DT-Q*.

For each candidate answer satisfying the query we still have to verify

rhetorical_agreement(subtree-DT-Q, A_i).

Once the answer A_i is selected and given to the user, user responds with C_i which in general case contains some clarification expressed in A_i and also an additional question part Q_i . The latter would then require an additional answer which should be added to A_i if it has been already computed.

The high-level view of the ***search algorithm*** that supports the dialogue is as follows:

1) Build *DT-Q*;
2) Split *DT-Q* into parts $Q_1, Q_2, \ldots$ to correspond to $A_1, A_2, \ldots$;
3) Form search query for A_1 from Q_1 in the form *RST-relation [phrase]* ;
4) Run the search against the set of dialogue utterances and obtain the list of candidate answers for the first step $A_{1\text{candidate}}$;
5) Build *DT-*$A_{1\text{candidate}}$ for each candidate and approve/reject each based on *rhetorical_agreement (DT–Q, DT-*$A_{1\text{candidate}}$*).* Select the best candidate A_1;
6) Respond to the user with the selected A_1 and receive C_1;
7) Form search query for A_2 from Q_1&C_1;
8) Repeat steps 4) and 5) for A_2 , respond to the user with the selected A_2 and receive C_2;
9) Conclude the session or switch to a human agent

Hence the dialogue management problem can be formulated as a search with constraints on DTs and can be implemented via traditional search engineering means plus discourse parsing, when an adequate set of chat logs is available. Discourse-tree based dialogue management does not cover all possibilities of assuring smooth dialogue flows but provides a plausible mechanism to select suitable utterances from the available set. It allows avoiding solving NL generation problem for dialogues that is a source of a substantial distortion of conversation flow and a noise in meaning of utterances.

In this paper we suggested a mechanism to build a dialogue structure where the first utterance formulated a detailed *Q* requiring some knowledge and explanation. If this *Q* is detailed enough the chat bot can attempt to find a sequence of answers to address all issues raised in *Q*.

3.5 Clarification-driven Dialogue

Some goals related to dialogue management do not need to be achieved via DTs. For example, an implementation of *clarification* feature can be hard-coded and does not require specific RRs. When a user asks a broad question, the chat bot forms topics for this user to choose from. Once such a topic is selected, the full answer is provided.

We have built an algorithm for mapping $Q \rightarrow A_i$. When multiple valid candidate answers for each *i* obtained, a clarification procedure is needed to have the user selected A_{ij} from the set A_i of answers or recommendations valid from both relevance and rhetorical agreement perspectives. We now update step 6) of the search algorithm which requests clarification when multiple suitable answers are available:

6) Respond to the user with the list of choices selected $\{A_{1s} | s=0..n\}$ and receive clarification result with possibly additional constraint request C_1 (which includes the choice A_{1j})

4 Evaluation of Dialogue Construction

4.1 Example from Car Repair Domain

As we proceed to evaluation, we will share an example of a dialogue from the evaluation domain of recommendations on car repair. Key characteristics of this domain are an importance of accurate recommendation, adequate dialogue structure, lesser bios than in other domains such as political, and an availability of a complete and accurate coverage of a broad spectrum of car problems. We provide an example of a sample dialogue:

Q_1: *I have checked the spark plugs, replaced the battery, also replaced ignition coils and the solenoid, but I still could not start my car.*

A_1: *Is the engine getting spark and fuel? Will it start for a couple seconds using starting fluid?*

Q_2: *It did at first but not anymore. What should I do?*

A_2 : *Check fuel pressure. But if no pressure, check the fuel pump for possible fault. If you have fuel pressure, then using a node light make sure you have injector pulse. The light should blink when cranking.*

If light is not blinking does the check engine light come on with key on and engine off? If no light then look for a bad main relay or no power - to the engine control module.

In this example, since Q_1 includes rhetorical relation of *Contrast* (something is done correctly but still not working), detailed answer should include *Contrast, Condition,* or other RR that would address *Contrast* in Q. A certain recommended action is performed either correctly or incorrectly and both these cases need to be addressed.

4.2 Dataset and Results

We formed a dataset of 9300 Q/A pairs related to car repair recommendations from www.2carpros.com. These pairs were extracted from dialogues as first and second utterance, so that the question is 7 – 15 keywords and answer is 3 to 6 sentences. This resource was obtained to train a dialogue support system but it also proved to be useful to evaluate search. The dataset is available online [2] in our GitHub.

To automate the relevance assessment, we considered the dialogue built *correctly* if an *actual* dialogue from the dataset is formed, given the first Q as a seed. Otherwise, if the sequence of utterances does not occur in the dataset, we consider it to be *incorrect*. There are some deficiencies of this approach since some actual dialogs are illogical and some synthetic dialogues built from distinct ones can be plausible, but it allows avoiding a manual tagging and construction of dialogues. The number of formed answers is limit to three: once initial Q is given, the system forms A_1, a set of A_{2i} and A_{3j}. A_1 is followed by the actual C_1 from the dialogue Q, so the proper A_2 needs to be selected. Analogously, once actual C_2 (if applicable) is provided, proper A_3 needs to be selected.

As a first baseline approach, we selected dialogue construction based on keyword similarity only, without taking into account a dialogue flow by considering a DT-Q. As a second baseline approach, we augment keyword similarity with linguistic relevance by computing maximal common sub-parse trees between the Q and A_i (Galitsky, 2013; Galitsky et al., 2013).

For the selected dataset, baseline approach is capable of building correct scenarios in the cases when similar keywords or similar linguistic phrases deliver the only dialogue scenario that is correct. On the contrary, *DT-Q* dialogue formation does not always succeed because some scenarios deviate from actual ones in the training set, although they are still plausible. Hence we see 10 and 5% improvement over the first and second baselines respectively for a basic, single-step scenario (Table 1).

Dialog type	Q-A	Q-A_1-C_1	Q-A_1-C_1-A_2	Q-A_1-C_1-A_2-C_2-A_3
Baseline 1	62.3±4.5	60.2±5.6	58.2±5.0	52.5±5.7
Baseline 2	67.0±4.8	63.8±4.8	57.3±5.3	55.6±5.9
DT-Q dialog formation	**72.3±5.6**	**70.3±4.9**	**65.1±5.5**	**65.9±5.7**

Table 1: Correctness of dialogue construction

As scenario becomes more complex, the chance that the proper scenario is selected by topic relevance decreases. At the same time, overall scenario formation complexity increases, and therefore an error rate for *DT-Q* approach increases as well. For the most complex, 3-step dialogue scenarios, *DT-Q* approach exceeds the baselines by 13 and 10% respectively.

5 Conclusions

In this paper we discovered that a dialogue structure could be built from the discourse tree of an initial question. This structure is built on top of the default conversational structure implementing such features as clarification, personalization or recommendation. For personalization, for a user query, the customer support chat bot system reduces the list of resolution scenarios based on what information is available for the given user. Chat bot recommendation scenario proposes a solution to a problem by finding the one accepted by users similar to the current one. Clarification, personalization and recommendation scenario covers only a small portion of plausible customer support scenarios. Discourse analysis of dialogues support dialogue scenario management in a universal way, for a broad range of available text fragments and previously accumulated responses.

Acknowledgments

The work of Dmitry Ilvovsky was supported by RFBR grants 16-29-12982 and 16-01-00583 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'.

[2] https://github.com/bgalitsky/relevance-based-on-parse-trees/blob/master/examples/CarRepairData_AnswerAnatomyDataset2.csv.zip

References

Wilks, Y. A. (Ed.). 1999. *Machine conversations.* Kluwer.

Mann, William 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.

Joty, Shafiq R, 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.

Mihai Surdeanu, Thomas Hicks, and Marco A. Valenzuela-Escarcega. 2015 *Two Practical Rhetorical Structure Theory Parsers.* NAACL HLT.

Chali, Y. Shafiq R. Joty, and Sadid A. Hasan. 2009. *Complex question answering: unsupervised learning approaches and experiments.* J. Artif. Int. Res. 35, 1 (May 2009), 1-47.

Boris Galitsky. 2017. *Discovering Rhetorical Agreement between a Request and Response.* Dialogue & Discourse 8(2) 167-205.

B Galitsky, D Ilvovsky, SO Kuznetsov, F Strok. 2013. *Matching sets of parse trees for answering multi-sentence questions*. RANLP-2013.

Boris Galitsky. 2013. *Machine Learning of Syntactic Parse Trees for Search and Classification of Text.* Engineering Application of Artificial Intelligence.