

Discovering Rhetorical Agreement between a Request and Response

Boris Galitsky

BORIS.GALITSKY@ORACLE.COM

Oracle Corp.

Redwood Shores CA USA

Editor: Barbara Di Eugenio

Submitted 11/2016; Accepted 12/2017; Published online 12/2017

Abstract

To support a natural conversation flow between humans and automated agents, the rhetorical structure of each message must be analyzed. We classified pairs of text paragraphs as either appropriate or inappropriate for one to follow the other based on considerations of both topic and communicative discourse. To represent a multi-sentence message with respect to how it should follow a previous message in a conversation or dialogue, we built an extension of a discourse tree. An extended discourse tree is based on a discourse tree for RST relations, with labels for communicative actions and additional arcs for anaphora and ontology-based relations for entities. We refer to such trees as communicative discourse trees (CDTs). We explored the syntactic and discourse features indicative of correct versus incorrect request-response or question-answer pairs. Two learning frameworks were used to recognize such correct pairs: deterministic, nearest-neighbor learning of CDTs as graphs and tree kernel learning of CDTs, in which a feature space of all CDT sub-trees is subject to SVM learning. We formed the positive training set from correct pairs obtained from Yahoo Answers, social networks, corporate conversations (including Enron emails) customer complaints and interviews by journalists. The corresponding negative training set was artificially created by attaching responses for different inappropriate answers that nevertheless cover the topics of questions. The evaluation showed that it is possible to recognize valid pairs in 70% of cases in the domains of weak request-response agreement and 80% of cases in the domains of strong agreement. Recognition of such pairs is essential to support automated conversations. These accuracy rates are comparable to the benchmark task of classifying discourse trees as either valid or invalid. They are also comparable to the classification of multi-sentence answers in factoid question-answering systems. We conclude that learning rhetorical structures in the form of CDTs is a key source of data to support answering complex questions.

1 Introduction

In recent years, the development of chatbots for answering questions and performing user requests has become very popular. A broad range of relevant technologies, including compositional semantics, have been developed to support these systems in the context of simple, short queries and replies. The accuracy of discourse parsing results has dramatically increased. Rhetorical parsers are now capable of building discourse structures for longer queries, requests and answers (Subba and Di Eugenio 2009). At present, the issue of how a question-answering,

dialogue management, or recommendation system (Galitsky 2013) can leverage rich discourse-related information in a structural form has not yet been extensively addressed.

During the last two decades, research in the field of dialogue systems has experienced increasing growth (Wilks 1999, Hiraoka et al., 2013). A number of formal systems representing various aspects of dialogue have been proposed (Traum and Hinkelman 1992, Blaylock et al., 2003, Popescu-Belis 2005, Popescu 2007, Visser et al., 2014). However, the design and optimization of these systems does not entail simply combining language processing systems such as parsers, part-of-speech taggers, intention models, rhetorical components and natural language generation systems. It also requires the development of dialogue strategies, considering at minimum the performances of these systems, the nature of the task (such as form-filling, tutoring, robot control, information and advice requests, social promotion or database search/browsing), and user behavior, such as cooperativeness and expertise (Aina et al., 2017). Due to the great variability in these factors, deploying manual, handcrafted dialogue designs is very difficult. For these reasons, statistical machine learning methods to support dialogue have been a leading focus of research for the last decade.

A request can have an arbitrary rhetorical structure as long as the topic of this request or question is clear to its recipient. A response on its own can also have an arbitrary rhetorical structure. However, these structures should be correlated when the response is appropriate to the request. In this study, we focused on the computational measurement of the agreement between the logical rhetorical structure of a request or question and that of the corresponding response or answer. We formed a number of representations for a request-response (RR) pair, learned them and solved an RR classification problem, identifying a pair as either valid (a correct answer or response) or invalid pairs.

Traditionally, computational models of communicative discourse are based on analyses of speaker intent (Allen and Perrault, 1980; Grosz and Sidner, 1986; Heller et al., 2013). In recent years, deep learning based models for dialogue learning have also become popular (Zhae et al., 2017). A requester has certain goals, and communication results from a planning process to achieve these goals. The requester will form intentions based on these goals and then act on these intentions, producing utterances. Upon hearing the utterance, the responder will then reconstruct a model of the requester’s intentions. However, this family of approaches is limited to providing an adequate account of adherence to discourse conventions in dialogue.

In addition to the analysis of intent, the notion of grounding has proved to be fruitful in modeling dialogues. Language needs grounding in the nonlinguistic world and in the practices of language users. This grounding is formed and controlled in the course of dialogue through conversational grounding (Schlangen, 2016), which is the interactive process through which interlocutors form an understanding of each other, grounding justification (the ability to explain and provide reasons for each agents’ language use), and grounding adaptation (the ability to accept corrections and modify how language is employed).

When answering a question formulated as a phrase or a sentence, the answer must address the *topic* of the question. Given an initial utterance as an explicit or implicit question, its answer is expected not only to maintain a topic but also to match the *generalized epistemic state* of this utterance. For example, when a person is looking to sell an item with features, the search results should not only contain these features but also indicate intent to buy. When a person is looking to share knowledge about an item, the search results should contain an intent to receive a recommendation. When a person asks for an opinion about a topic, the response should be to share an opinion about this topic instead of expressing another request for an opinion. Modern dialogue management and automated email-answering systems have achieved good accuracy with

regard to maintaining the topic, but maintaining the communication discourse is a much more difficult problem.

The syntactic structure of a simple question is correlated with that of an answer. This structure is helpful for finding the best answer in the passage re-ranking problem. It has been shown that using syntactic information to improve search relevance is helpful in addition to keyword frequency (TF*IDF) analysis and other keyword statistical methods, such as LDA (Blei et al., 2003). Selecting a most suitable answer, not only through keywords but also by judging how the syntactic structure of a question, including a focus on Wh-words, is reflected in an answer, has been proposed (Moschitti and Quarteroni 2011; Galitsky 2013). Following along the lines of these studies, we have adapted this consideration at the phrase and sentence levels and applied it to the level of discourse.

To represent the linguistic features of text, we used the two following sources:

1. *Rhetorical relations* between the parts of the sentences, obtained as a *discourse tree*. We relied on Rhetorical Structure Theory (RST, Mann and Thompson 1988) and deployed rhetorical parsers (Joty et al., 2013; Surdeanu et al., 2015) to build these discourse trees.
2. *Speech acts and communicative actions*, obtained as verbs from the VerbNet resource (verb signatures with instantiated semantic roles). These were attached to rhetorical relations as labels for the arcs of communicative discourse trees.

It turns out that having only 1) or only 2) is insufficient for recognizing correct RR pairs. However, the combination of these sources is sufficient.

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 (DTs). The leaves of a DT correspond to EDUs, the contiguous atomic text spans. Adjacent EDUs are connected by coherence 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.

The goal of this research was to extend the notion of question/answer relevance to the *rhetorical* relevance of general request/response pairs for broader dialogue support.

We now proceed to an example for an agreement between a question and answer. For the question

“*What does The Investigative Committee of the Russian Federation do*” there are two answers:

- 1) *Mission statement*. “The Investigative Committee of the Russian Federation is the main federal investigating authority which operates as Russia's Anti-corruption agency and has statutory responsibility for inspecting the police forces, combating police corruption and police misconduct, is responsible for conducting investigations into local authorities and federal governmental bodies.”
- 2) *An answer from the web*. “Investigative Committee of the Russian Federation is supposed to fight corruption. However, top-rank officers of the Investigative Committee

of the Russian Federation are charged with creation of a criminal community. Not only that, but their involvement in large bribes, money laundering, obstruction of justice, abuse of power, extortion, and racketeering has been reported. Due to the activities of these officers, dozens of high-profile cases including the ones against criminal lords had been ultimately ruined” (CrimeRussia 2016).

The choice of answers depends on context. Rhetorical structure allows differentiation between “official,” “politically correct,” template-based answers and “actual,” “raw,” “reports from the field,” “controversial” ones (Fig. 1 a and b). Sometimes, the question itself can give a hint about which category of answers is expected. If a question is formulated as a factoid or definitional question without a second meaning, then the first category of answers is suitable. Otherwise, when a question has the meaning, “tell me what it *really* is,” the second category is appropriate. In general, if we can extract a rhetorical structure from a question, it is easier to select a suitable answer that would have a similar, matching, or complementary rhetorical structure.

The discourse trees of an official answer are based on *elaboration* and *joints*, which are neutral in terms of any controversy that a text might contain (Fig. 1a). At the same time, the raw answer includes the *contrast* relation. This relation refers to the contrast between what an agent is expected to do and what the agent was discovered to have done.

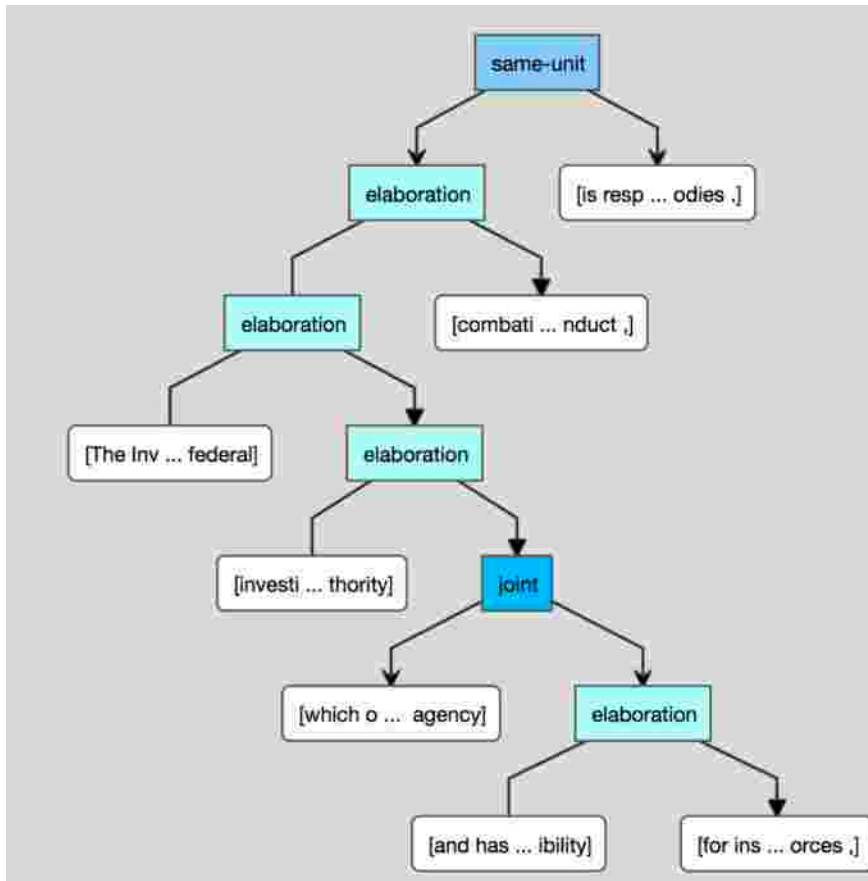


Fig. 1a: DT for the official answer

In the future (but not currently), discourse parsers should be capable of differentiating between

“What does this entity do” and “What does this entity *really* do” by identifying a contrast relation between ‘do’ and ‘really do.’ After such a relation is established, it would be easier to make a decision as to whether an answer with or without contrast is suitable. Hence, when rhetorical parsing improves, the same DT learning machinery would deliver more accurate rhetorical agreement results.

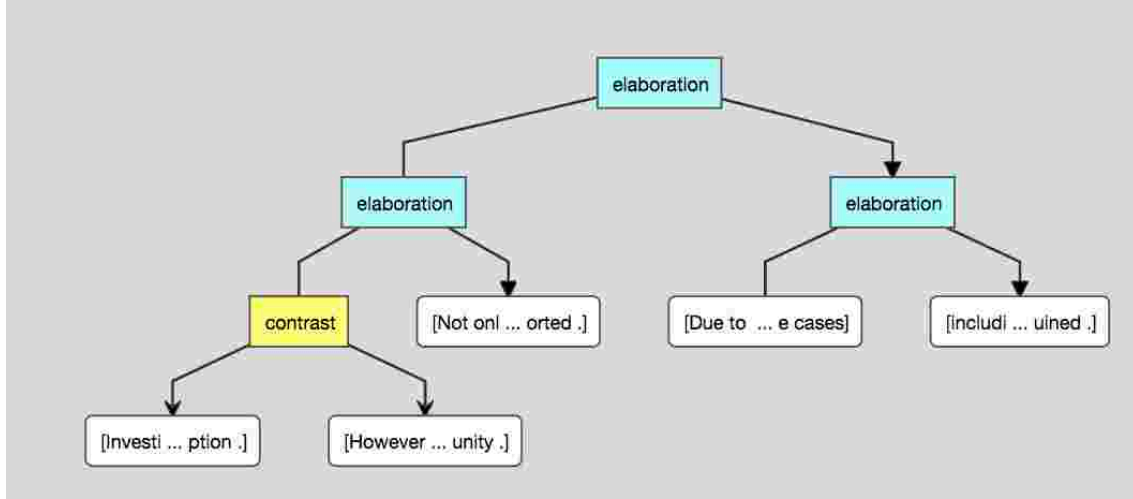


Fig. 1b: DT for the raw answer (from the web)

We formulated the main problem of this study to classify pairs of RR texts as correct or incorrect. This problem can be formulated with or without considering relevance, which we intend to treat orthogonally to how the rhetorical structure of a request agrees with the rhetorical structure of a response. Rhetorical agreement may be present or absent in an RR pair, and the same applies to the relevance agreement. Some methods of measuring rhetorical agreement include considerations of relevance agreement whereas others do not.

The idea of measuring the similarity between question-answer pairs for question-answering instead of question-answer similarity turned out to be insightful (Moschitti and Quarteroni, 2011). The classifier for correct versus incorrect answers processes two pairs at a time, $\langle q_1, a_1 \rangle$ and $\langle q_2, a_2 \rangle$, and compares q_1 with q_2 and a_1 with a_2 , producing a combined similarity score. This comparison allows the classifier to determine whether an unknown question/answer pair contains a correct answer by assessing its distance from another question/answer pair with a known label. In particular, an unlabeled pair $\langle q_2, a_2 \rangle$ is processed such that, rather than “guessing” correctness based on words or structures shared by q_2 and a_2 , both q_2 and a_2 are compared with their corresponding components, q_1 and a_1 of the labeled pair $\langle q_2, a_2 \rangle$, on the grounds of such words or structures. Because this approach targets a domain-independent answer classification, only the structural cohesiveness between a question and answer, not the ‘meaning’ of an answer, can be leveraged.

To form a training set for this classification problem, we included actual RR pairs in the positive dataset and arbitrary or low-relevance and -appropriateness RR pairs in the negative dataset. For the positive dataset, we selected various domains with distinct acceptance criteria for which an answer or response was suitable for the question. Such acceptance criteria are low for community question-answering, automated question-answering, automated and manual customer support systems, social network communications and writing by individuals such as consumers about their experience with products such as reviews and complaints. RR acceptance criteria are higher in scientific texts, professional journalism, health and legal documents in the form of

FAQs, and in professional social networks, such as stackoverflow.com.

2 Communicative Discourse Trees

Communicative discourse trees (CDTs) are designed to combine rhetorical information in the form of DT with speech act structures using arcs labeled with expressions for communicative actions. These expressions are logical predicates expressing the agents involved in the respective speech acts and the arguments of their communicative actions. The arguments of logical predicates are formed in accordance with their respective semantic roles as proposed by a framework such as VerbNet (Kipper et al., 2008, Palmer 2009). The purpose of adding these labels is to incorporate the speech act-specific information into DTs so that their learning occurs over a richer feature set than just rhetorical relations and the syntax of elementary discourse units (EDUs). The objective here is to incorporate all information about how an author's thoughts are organized and communicated irrespective of the subjects of these thoughts.

Our second example of a rhetorical agreement in a conversation is a dispute between three parties concerning the cause of the downing of Malaysia Airlines Flight 17 (Wikipedia 2016). We built an RST representation of the arguments being communicated and observed if a discourse tree is capable of indicating whether a paragraph communicates both a claim and an argumentation that backs it up. We then explored what needs to be added to the DT representation so that it is possible to judge whether it expresses an argumentation pattern or not. Surdeanu et al.'s (2015) computation and visualization system for DT was used.

Three conflicting agents, *Dutch investigators*, *The Investigative Committee of the Russian Federation*, and *the self-proclaimed Donetsk People's Republic*, exchanged their opinions on the matter. It is a controversial conflict in which each party does all it can to blame its opponent. To sound more convincing, each party does not just produce its claim but formulates it in such a way as to rebuff the claims of its opponent. To achieve this goal, each party attempts to match the style and discourse of the opponents' claims.

"Dutch accident investigators say that evidence points to pro-Russian rebels as being responsible for shooting down plane. The report indicates where the missile was fired from and identifies who was in control of the territory and pins the downing of MH17 on the pro-Russian rebels." (Fig. 2a)

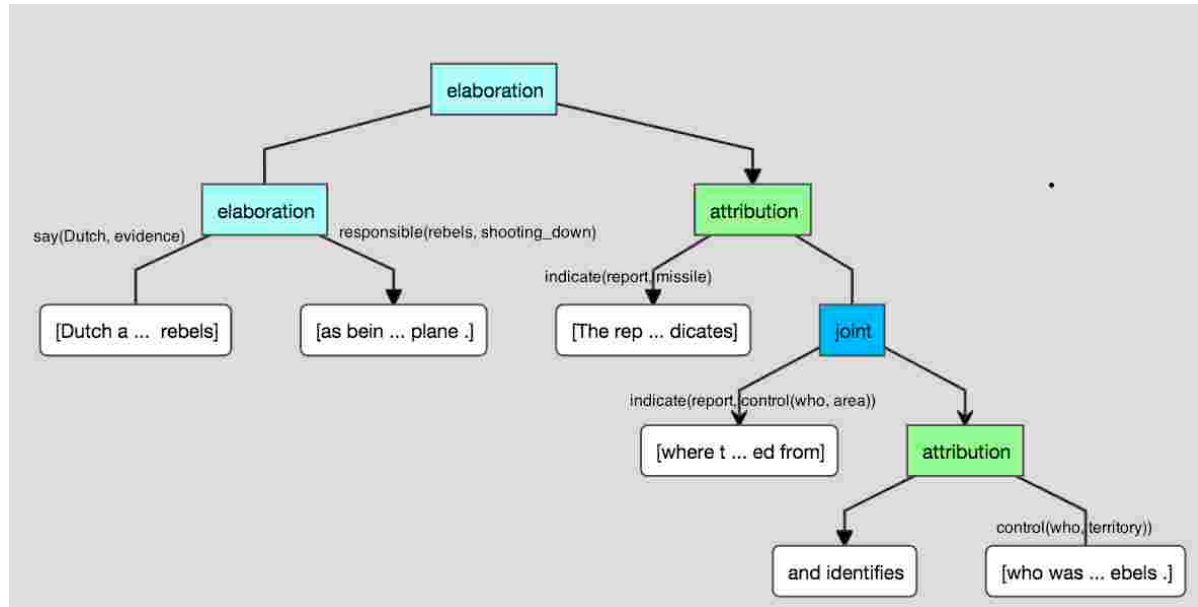


Fig. 2a: The claim of the first agent, *Dutch accident investigators*.

The regular nodes of CDTs are rhetorical relations, and the terminal nodes are the elementary discourse units (phrases, sentence fragments) that are the arguments of these relations. Certain arcs of CDTs are labeled with the expressions for communicative actions, including the actor agent and the argument of these actions (what is being communicated). For example, the nucleus nodes for elaboration relation (on the left) are labeled with *say (Dutch, evidence)*, and the satellites, with *responsible (rebels, shooting_down)*. These labels are not intended to express that the arguments of EDUs are *evidence* and *shooting_down* but instead to match this CDT with others to find the similarity between them. In this case, simply linking these communicative actions by a rhetorical relation but not providing information of communicative discourse would be too limited to represent a structure of what is being communicated and how. Requiring an RR pair to have the same or coordinated rhetorical relations is too weak; thus, an agreement of the CDT labels for arcs on top of matching nodes is required.

“The Investigative Committee of the Russian Federation believes that the plane was hit by a missile, which was not produced in Russia. The committee cites an investigation that established the type of the missile.” (Fig. 2b)

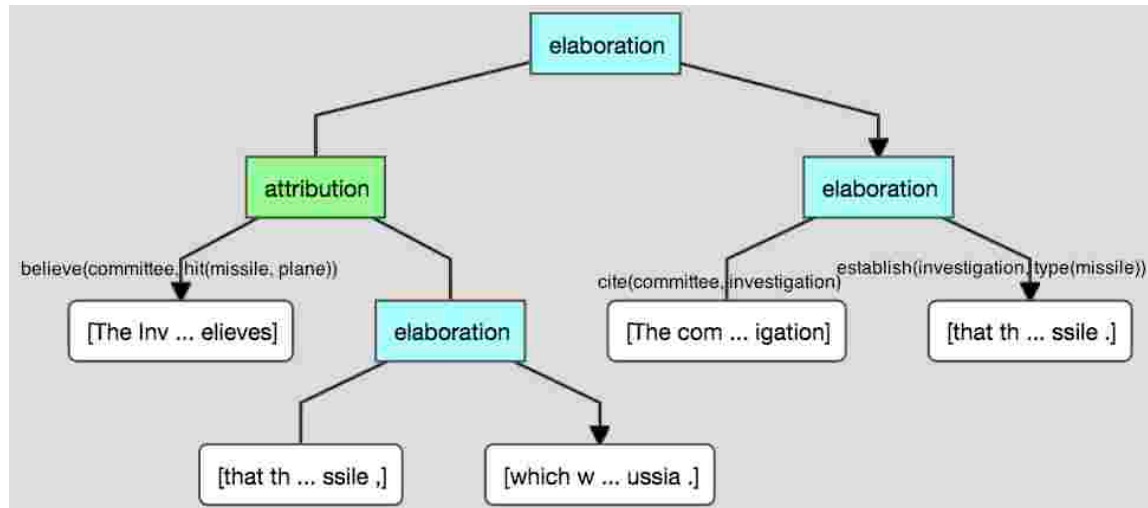


Fig. 2b: The claim of the second agent, the Committee.

“Rebels, the self-proclaimed Donetsk People's Republic, deny that they controlled the territory from which the missile was allegedly fired. It became possible only after three months after the tragedy to say if rebels controlled one or another town.” (Fig. 2c).

A response cannot be arbitrary. It has to talk about the same entities as the original text. It has to back up its disagreement with its estimates and sentiments about these entities, and about actions of these entities. We attempt to encode the structure of agreement between RR pairs via CDT in a domain-independent manner, only in the space of communication and its style, and the logical flow of a conversation irrespectively of the nature of these entities.

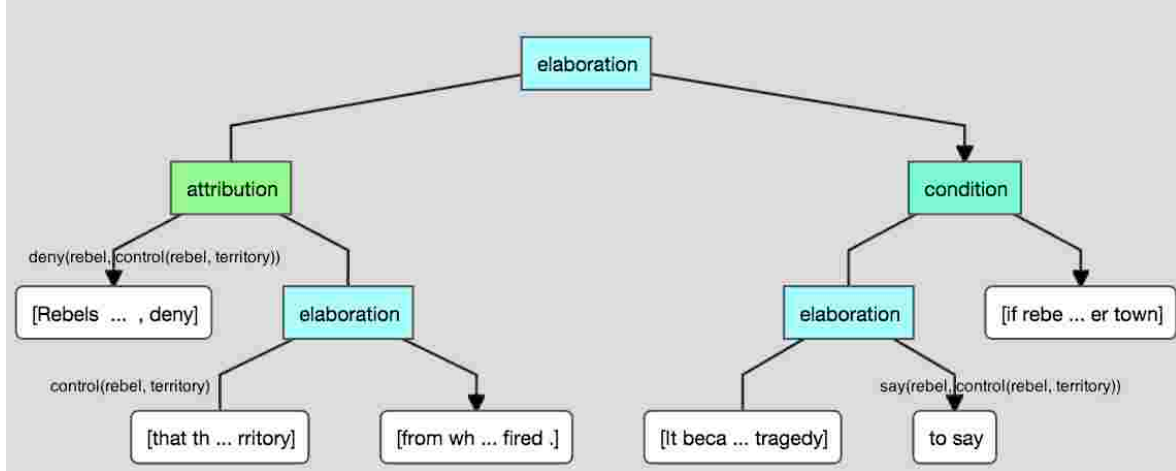


Fig. 2c: The claim of the third agent, *the rebels*.

What we see from this example is that the replies of the peers need to reflect, to mimic the communicative discourse of the first utterance, the seed. As a simple observation, since the first agent uses *Attribution* to communicate his claims, the other agents have to follow suit and either provide their own attributions or attack the validity of attribution of the proponent, or both. To capture a broad variety of features for how communicative structure of the seed message needs to be retained in consecutive messages, we will learn the pairs of respective CDTs.

To verify the RR agreement, discourse relations are necessary but insufficient, and speech acts (communicative actions) are necessary but insufficient as well. For the paragraph from the previous example, we need to know the discourse structure of interactions between agents, and what kind of interactions they are. We don't need to know domain of interaction (here, military conflicts), the topics of these interaction, what the entities are. but we need to take into account mental, domain-independent relations between them.

Towards the end of this section we give a formal definition of CDT. *CDT is a DT with labels for arcs which are the VerbNet expressions for verbs which are communicative actions. The argument of verbs are substituted from text according to VerbNet frames.* For the details of DTs we refer the reader to (Joty et al., 2016), and for VerbNet Frames – to the section on communicative actions below and then to (Kipper et al., 2008).

We conclude this section with a note that a CDT required to learn RR agreement is an extension of a traditional discourse tree. It allows us to make RST relations labeled with communicative actions. CDT is a reduction of what is called parse thicket (Galitsky et al., 2015), a combination of parse trees for sentences with discourse-level relationships between words and parts of the sentence in one graph (Fig. 3). The straight edges of this graph are syntactic relations, and curvy arcs – discourse relations, such as anaphora, same entity, sub-entity, rhetorical relation and communicative actions. This graph includes much richer information than just a combination of parse trees for individual sentences would. As well as CDTs, parse thickets can be generalized at the level of words, relations, phrases and sentences.

3 Representing rhetorical relations and communicative actions

To compute similarity between abstract structures, two approaches are frequently used:

- 1) represent these structures in a numerical space, and express similarity as a number. This is a statistical learning approach

- 2) use a structural representation, without numerical space, such as trees and graphs, and express similarity as a maximal common sub-structure. We refer to such operation as *generalization*. This is an inductive learning approach.

To conduct feature engineering, we will compare both these approaches in the domain of this study. The representation machinery and learning settings are different, but the classification accuracies can be compared.

3.1 Greedy representations for a Q/A pair

We now proceed to another examples of Q/A pair and its representation trying to involve as detailed linguistic information as possible. A greedy approach to representing linguistic information about this pair and a map from Q to A is shown in Fig. 3. We combine parse trees for sentences with pragmatic and discourse-level relationships between words and parts of the sentence in one graph, called *parse thicket*. The parse thicket for Q is shown as a connected graph on the top, and for A – on the bottom.

We complemented the edges for syntactic relations obtained and visualized with the Stanford NLP system (Manning et al., 2014). (Recasens et al., 2013 and Lee et al 2013) was used for coreference resolution. The arcs for pragmatic and discourse relations, such as anaphora, same entity, sub-entity, rhetorical relation and communicative actions correspondence between parse thickets for Q and A are manually drawn. Labels embedded into arcs denote the syntactic relations. Lemmas are written below the boxes for the nodes, and parts-of-speech are written inside the boxes.

This graph includes much richer information than just a combination of parse trees for individual sentences would. Navigation through this graph along the edges for syntactic relations as well as arcs for discourse relations allows to transform a given parse thicket into semantically equivalent forms for matching with other parse thickets, performing a text similarity assessment task. To form a complete formal representation of a paragraph, we attempt to express as many links as possible: each of the discourse arcs produces a pair of thicket phrases that can be a potential match.

Topical similarity between Q and A can be expressed as common sub-graphs of parse thickets. The higher the number of common graph nodes, the higher the similarity. For rhetorical agreement, the common sub-graph does not have to be large. However, rhetorical relations and communicative actions of the seed and response are correlated and a correspondence is required.

Our example for a pair of parse thickets for a question and its answer is obtained for the following:

Q: I just had a baby and it looks more like the husband I had my baby with. However it does not look like me at all and I am scared that he was cheating on me with another lady and I had her kid. This child is the best thing that has ever happened to me and I cannot imagine giving my baby to the real mom.

A: Marital therapists advise on dealing with a child being born from an affair as follows. One option is for the husband to avoid contact but just have the basic legal and financial commitments. Another option is to have the wife fully involved and have the baby fully integrated into the family just like a child from a previous marriage.

3.2 Communicative actions and their generalization

Arguments are mostly communicated in the mental world, and learning communicative actions is a key to expressing and understanding arguments. Computational verb lexicons are key to supporting acquisition of entities for actions, and a rule-based form to express their meanings. Verbs express the semantics of an event being described as well as the relational information among participants in that event, and project the syntactic structures that encode that information. Verbs, and in particular the ones for communicative actions, are also highly variable, displaying a rich range of semantic behaviors. Verb classification helps a learning systems to deal with this complexity by organizing verbs into groups that share core semantic properties.

VerbNet is one such lexicon, which identifies semantic roles and syntactic patterns characteristic of the verbs in each class and makes explicit the connections between the syntactic patterns and the underlying semantic relations that can be inferred for all members of the class. Each syntactic frame in a class has a corresponding semantic representation that details the semantic relations between event participants across the course of the event. VerbNet is a good source of information on verbs in general and communicative actions in particular.

Let us consider the verb *amuse*. There is a cluster of similar verbs that have a similar structure of arguments (semantic roles) such as *amaze*, *anger*, *arouse*, *disturb*, *irritate*, and other. The roles of the arguments of these communicative actions are as follows:

- ∞ Experiencer (usually, an animate entity)
- ∞ Stimulus
- ∞ Result

The frames (the classes of meanings differentiated by syntactic features for how this verb occurs in a sentence) are as follows (NP – noun phrase, N – noun, V – communicative action, VP – verb phrase, ADV - adjective). Below is a set of definitions for the verb *amuse* (Fig. 3a).

For this example, the information for the class of verbs *amuse* is available at <http://verbs.colorado.edu/verb-index/vn/amuse-31.1.php#amuse-31.1>

We now show how communicative actions are split into clusters (Table 1).

The purpose of defining the similarity of two verbs as an abstract verb-like structure is to support inductive learning tasks such as rhetorical agreement assessment. In statistical machine learning, similarity is expressed as a number. A drawback of this learning approach is that, by representing linguistic feature space as numbers, one loses the ability to explain the learning feature underlying a classification decision. After a feature has been expressed as a number and combined with other numbers, it is difficult to interpret it. In inductive learning, when a system performs classification tasks, it identifies a particular verb or verb-like structure that is determined to cause the target feature (such as rhetorical agreement). In contrast, the statistical and deep learning-based family of approaches simply delivers decisions without explanation. In statistical learning, the similarity between two verbs is a number. In inductive learning, it is an abstract verb with attributes shared by these two verbs (attributes present in one but absent in the other are not retained). The resultant structure of similarity computation can be subjected to further similarity computation with another structure of that type or with another verb. For verb similarity computation, it is insufficient to indicate only that two verbs belong to the same class: all common attributes must occur in the similarity expression.

NP V NP

Example: "The teacher amused the children."

Syntax: Stimulus V Experiencer

Clause:

amuse(Stimulus, E, Emotion, Experiencer):-
 cause(Stimulus, E),
 emotional_state(result(E), Emotion, Experiencer).

NP V ADV-Middle

Example: "Small children amuse quickly."

Syntax: Experiencer V ADV

Clause:

amuse(Experiencer, Prop):-
 property(Experiencer, Prop), adv(Prop).

NP V NP-PRO-ARB

example "The teacher amused."

syntax Stimulus V

amuse(Stimulus, E, Emotion, Experiencer):-
 cause(Stimulus, E),
 emotional_state(result(E), Emotion, Experiencer).

NP.cause V NP

example "The teacher's dolls amused the children."

syntax Stimulus <+genitive> ('s) V Experiencer

amuse(Stimulus, E, Emotion, Experiencer):-
 cause(Stimulus, E),
 emotional_state(during(E), Emotion, Experiencer).

NP V NP ADJ

example "This performance bored me totally."

syntax Stimulus V Experiencer Result

amuse(Stimulus, E, Emotion, Experiencer):-
 cause(Stimulus, E),
 emotional_state(result(E), Emotion, Experiencer),
 Pred(result(E), Experiencer).

Fig. 3a: Definitions for the verb *amuse*

Verbs with Predicative Complements	<i>appoint, characterize, dub, declare, conjecture, masquerade, orphan, captain, consider, classify</i>
Verbs of Perception	<i>see, sight, peer</i>
Verbs of Psychological State	<i>amuse, admire, marvel, appeal</i>
Desire	want, long
Judgment Verbs	Judge
Assessment	<i>assess, estimate</i>
Verbs of Searching	<i>hunt, search, stalk, investigate, rummage, ferret</i>
Verbs of Social Interaction	<i>correspond, marry, meet, battle</i>
Verbs of Communication	<i>transfer(message), inquire, interrogate, tell, manner(speaking), talk, chat, say, complain, advise, confess, lecture, overstate, promise</i>
Avoid	<i>Avoid</i>
Measure	<i>register, cost, fit, price, bill</i>
Aspectual	<i>begin, complete, continue, stop, establish, sustain</i>

Table 1: VerbNet classes of the verbs for communicative actions

Hence similarity between two communicative actions A_1 and A_2 is defined as an abstract verb which possesses the features which are common between A_1 and A_2 . We first provide an example of the generalization of two very similar verbs:

$agree \wedge disagree = verb(Interlocutor, Proposed_action, Speaker),$

where *Interlocutor* is the person who proposed the *Proposed_action* to the *Speaker* and to whom the *Speaker* communicates their response. *Proposed_action* is an action that the *Speaker* would perform if they were to accept or refuse the request or offer, and the *Speaker* is the person to whom a particular action has been proposed and who responds to the request or offer made.

When verbs are not that similar, a subset of the arguments remain:

$agree \wedge explain = verb(Interlocutor, *, Speaker).$

Further examples of generalizing verbs and communicative actions are available in (Galitsky et al., 2009).

The main observation concerning communicative actions in relation to finding text similarity is that their arguments need to be generalized in the context of these actions and that they should not be generalized with other “physical” actions. Hence, we generalize the individual occurrences of communicative actions together with their arguments. We also generalize sequences of communicative actions representing dialogs against other such sequences of similar dialogs. This way we represent the meaning of an individual communicative action as well as the dynamic discourse structure of a dialogue (in contrast to its static structure reflected via rhetorical relations). The idea of generalization of compound structural representation is that generalization

happens at each level. The verb itself of a communicative action is generalized with another verb, and its semantic roles are generalized with their respective semantic roles.

Generalization of communicative actions can also be thought of from the standpoint of matching the verb frames. The communicative links reflect the discourse structure associated with participation (or mentioning) of more than a single agent in the text. The links form a sequence connecting the words for communicative actions (either verbs or multi-words implicitly indicating a communicative intent of a person).

For a communicative action, we distinguish an agent, one or more arguments being acted upon, and the phrase describing the features of this action. We define communicative action as a function of the form $verb(agent, argument, cause)$, where *verb* characterizes some type of interaction between involved *agents* (e.g., *explain*, *confirm*, *remind*, *disagree*, *deny*, etc.); *argument* refers to the information transmitted or object described, and *cause* refers to the motivation or explanation for the subject.

A scenario (labeled directed graph) is a sub-graph of a parse thicket $G=(V, A)$, where

$V=\{action_1, action_2, \dots, action_k\}$ is a finite set of vertices corresponding to communicative actions, and A is a finite set of labeled arcs (ordered pairs of vertices), classified as follows:

- ∞ Each arc $(action_i, action_j) \in A_{sequence}$ corresponds to a temporal precedence of two actions (v_i, ag_i, s_i, c_i) and (v_j, ag_j, s_j, c_j) referring to the same argument (that is, $s_i = s_j$) or different argument.

Each arc $(action_i, action_j) \in A_{cause}$ corresponds to an logical argumentation attack relationship (Gabbay and Garcez, 2009) between $action_i$ and $action_j$ indicating that the cause of $action_i$ is in conflict with the subject or cause of $action_j$.

Subgraphs of parse thickets which are associated with scenarios of interaction between agents have some distinguishing features (Galitsky et al., 2009):

- 1) all vertices are ordered in time, so that there is one incoming arc and one outgoing arc for all vertices (except the initial and terminal vertices);
- 2) for $A_{sequence}$ arcs, at most one incoming and only one outgoing arc are admissible;
- 3) for A_{cause} arcs, there can be many outgoing arcs from a given vertex, as well as many incoming arcs. The vertices involved may be associated with different agents or with the same agent (i.e., when he contradicts himself). To compute similarities between parse thickets and their communicative action – induced subgraphs – the sub-graphs of the same configuration with similar labels of arcs and strict correspondence of vertices need to be analyzed (Galitsky and Kuznetsov, 2008).

Analyzing the communicative actions' arcs of a parse thicket, one can find implicit similarities between texts. Given two texts T_1 and T_2 , we can generalize:

1. one communicative actions with its argument from T_1 against another communicative action with its argument from T_2 (communicative action arc is not used) ;
2. a pair of communicative actions with their arguments from T_1 against another pair of communicative actions from T_2 (communicative action arcs are used) .

In our example in Fig. 3 we have the former case:

cheating(husband, wife, another lady)
 \wedge
avoid(husband, contact(husband, another lady))

This generalization gives us *communicative_action(husband, *)* which introduces a constraint on A in the form that if a given agent (= *husband*) is mentioned as an argument of CA in Q, he/she should also be an argument of (possibly, another) CA in A.

To handle meaning of words expressing the arguments of CAs, we apply compositional semantics (Mikolov et al., 2011, Mikolov et al., 2015) models. To compute generalization between the arguments of communicative actions, we use the following rule:

if $argument_1 = argument_2$, $argument_1 \wedge argument_2 = \langle argument_1, POS(argument_1), 1 \rangle$. Here *subject* remains and score is 1.

Otherwise, if they have the same part-of-speech (POS),

$argument_1 \wedge argument_2 = \langle *, POS(argument_1), word2vecDistance(argument_1 \wedge argument_2) \rangle$. ‘*’ denotes that lemma is a placeholder, and the score is a word2vec distance between these words.

If POS’ are different, the generalization is an empty tuple. It cannot be further generalized.

As the reader can observe, generalization results can be further generalized with other arguments of communicative actions and with their generalizations.

Generalizing two different communicative actions is based on their attributes and is presented elsewhere (Galitsky et al., 2013).

3.3 Generalization for RST relations

Only RST arcs of the same type of relation (*presentation* relations, such as *antithesis*; *subject matter* relations, such as *condition*; and *multinuclear* relations, such as *list*) can be generalized. We use *N* for a nucleus or situation presented by this nucleus, and *S* for satellite or situation presented by this satellite, *W* for a writer, and *R*, for a reader (hearer). *Situations* are propositions, completed actions or actions in progress, or communicative actions, or states (including *beliefs*, *desires*, *approve*, *explain*, *reconcile* and others).

The generalization of two RST relations with the above parameters is expressed as

$$rst_1(N_1, S_1, W_1, R_1) \wedge rst_2(N_2, S_2, W_2, R_2) = (rst_1 \wedge rst_2)(N_1 \wedge N_2, S_1 \wedge S_2, W_1 \wedge W_2, R_1 \wedge R_2).$$

The texts in N_1, S_1, W_1, R_1 are subject to generalization as phrases.

The rules for $rst_1 \wedge rst_2$ are as follows:

- ∞ If $relation_type(rst_1) \neq relation_type(rst_2)$ then generalization is empty.
- ∞ Otherwise, we generalize the signatures of rhetorical relations as sentences (Iruskieta et al., 2015):

$$sentence(N_1, S_1, W_1, R_1) \wedge sentence(N_2, S_2, W_2, R_2).$$

For example, we apply generalization to the definitions of RST relations:

$$rst_background \wedge rst_enablement = (S \text{ increases the ability of } R \text{ to comprehend an element in } N) \wedge (R \text{ comprehending } S \text{ increases the ability of } R \text{ to perform the action in } N) = \textit{increase-VB the-DT ability-NN of-IN R-NN to-IN}.$$

Since the relations $rst_background \wedge rst_enablement$ are different, the RST relation part is empty. We then generalize the expressions that are the verbal definitions of the respective RST relations. For each word or a placeholder for a word such as an agent, we retain this word (with its POS) if it is the same in each input phrase or remove it if it is different between these phrases. The resultant expression can be interpreted as a common meaning between the definitions of two different RST relations, obtained formally. The reader is recommended to consult Galitsky et al., 2012 for further details of syntactic generalization.

A mapping between two essential rhetorical relations RST-Contrast is shown in the middle of Fig. 3. This mapping is important to demonstrate that if contrasting clauses occur in the question, they have to be addressed (most likely, via *contrast*) in the answer as well.

To compute the generalization between the expressions for contrast in Q and A, we draw a chart:

<i>I just had a baby</i>	$\leftrightarrow_{\text{RST-Contrast}}$	<i>it does not look like me</i>
^		
<i>husband to avoid contact</i>	$\leftrightarrow_{\text{RST-Contrast}}$	<i>have the basic legal and financial commitments</i>

The generalization gives us $\text{VP} \leftrightarrow_{\text{RST-Contrast}} \text{VP}$ which is read as “Both *Q* and *A* have contrast expressed via verb phrases”.

We can see that the VP of *A* does not have to be similar to the VP of *Q* but their rhetorical structure does. Not all phrases in *A* must match phrases in *Q*: those which do not match must be in certain rhetorical relations with those *A* phrases which are relevant to phrases in *Q*.

3.4 Representing a Request-Response chain

So far we considered the stand-alone R-R pairs. In this section we explore how these pairs can form a chain and how to represent its structure. Once we have a chain, a rhetorical agreement is expected to hold not only between consecutive members but also triples and four-tuples. For a text expressing a sequence of R-R pairs, we will compare the scenario representation with the discourse tree representation.

In the domain of customer complaints, request and response are present in the same text, from the viewpoint of a complainant. The customer complaint text needs to be split into request and response text portions to form the positive and negative dataset of pairs. For the purpose of evaluation, we combine all text for the proponent and all text for the opponent together. The first sentence of each paragraph below will form the *Request* part (which will include three sentences) and the second sentence of each paragraph will form the *Response* part (which will also include three sentences in this example).

Let us consider a scenario with communicative actions and its representation via DT and scenario graph (Fig. 5).

*I **explained** that my check bounced (I wrote it after I made a deposit). A customer service representative **accepted** that it usually takes some time to process the deposit.*

*I **reminded** that I was unfairly charged an overdraft fee a month ago in a similar situation. They **denied** that it was unfair because the overdraft fee was disclosed in my account information.*

*I **disagreed** with their fee and wanted this fee deposited back to my account. They **explained** that nothing can be done at this point and that I need to look into the account rules closer.*

A Discourse Tree and communicative scenario for this conflict dialogue is shown in Fig. 4. Judging by the DT, it is hard to see if this text is an interaction scenario or just some kind of description. The scenario graph on the bottom is a higher-level representation of discourse.

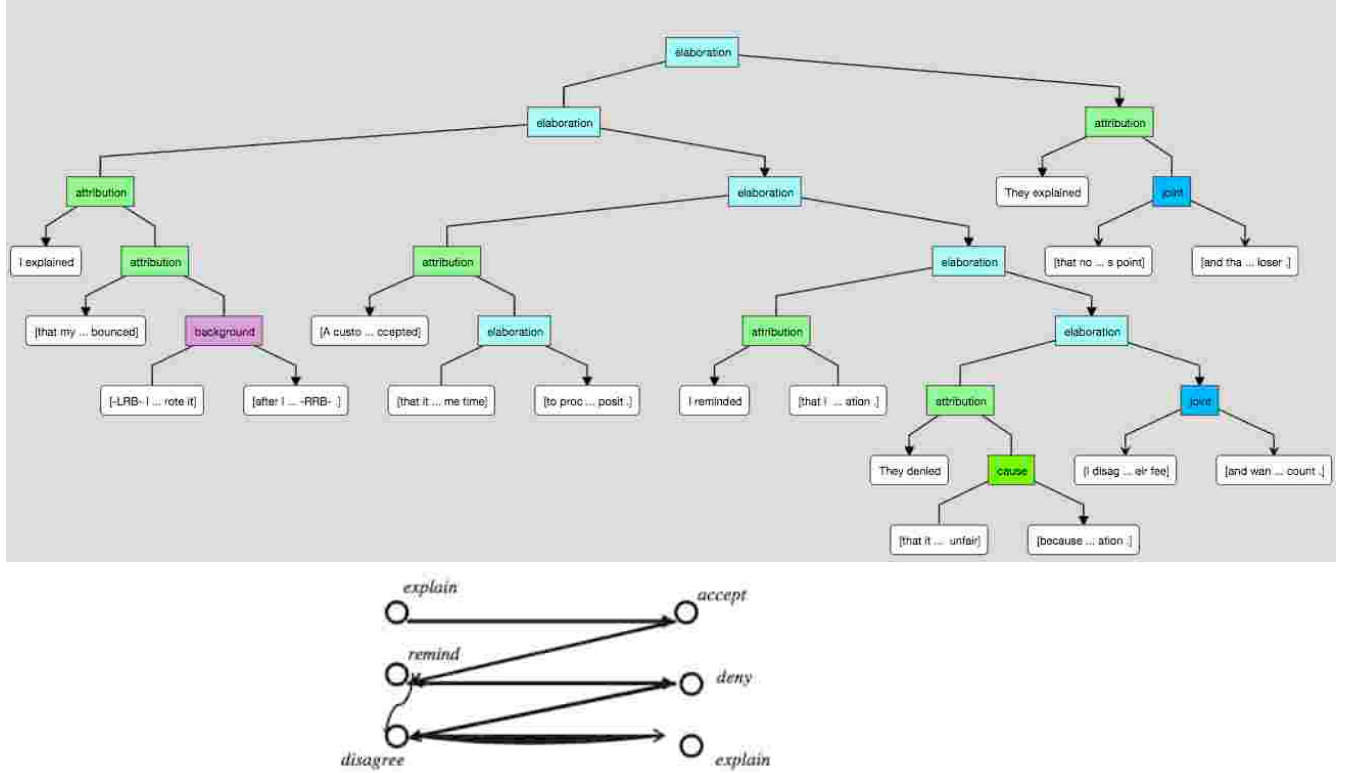


Fig. 4: Discourse tree on the top and scenario graph for communicative actions on the bottom. These two sources of discourse information complement each other.

4 Classification settings for Request-Response pairs

In a conventional search approach, as a baseline, RR 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 2015). Also, search can be formulated as a passage re-ranking problem in a machine learning framework. The feature space includes RR pairs as elements, and a separation hyper-plane splits this feature space into correct and incorrect pairs. Hence, a search problem can be formulated in a local way, as similarity between *Req* and *Resp*, and in a global, learning way, via similarity between RR pairs.

We take these groups of methods further towards discourse level analysis. To measure the RR match there are the following classes of methods:

1. Extract features for *Req* and *Resp* and compare them as a feature count. Introduce a scoring function such that a score would indicate a class (low score for incorrect pairs, high score for correct ones);
2. Compare representations for *Req* and *Resp* against each other, and assign a score for the comparison result. Analogously, the score will indicate a class;
3. Build a representation for a pair *Req* and *Resp*, $\langle \text{Req}, \text{Resp} \rangle$ as elements of the training set. Then perform learning in the feature space of all such elements $\langle \text{Req}, \text{Resp} \rangle$.

To form a $\langle \text{Req}, \text{Resp} \rangle$ object we combine DT(*Req*) with DT(*Resp*) into a single tree with the root RR (Fig. 5). We then classify such objects into correct (with high agreement) and incorrect (with low agreement).

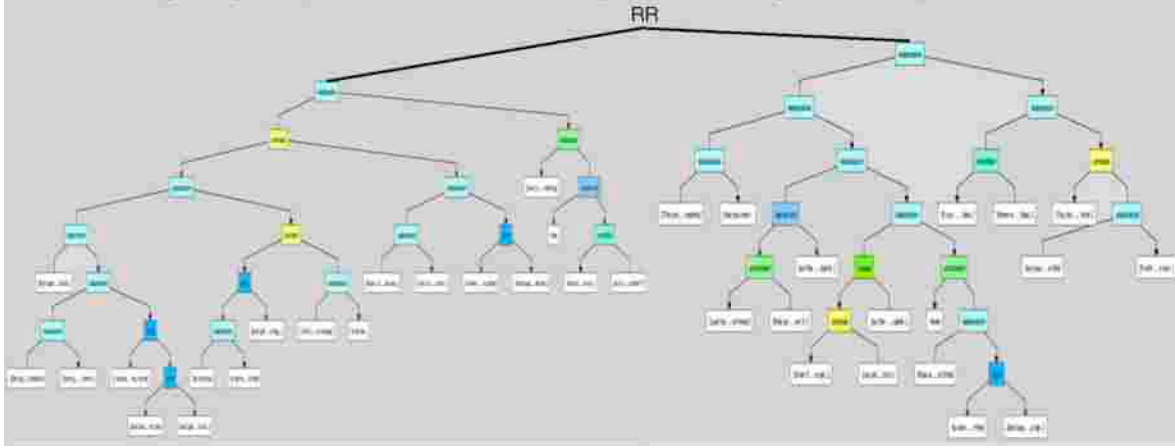


Fig. 5: Forming the Request-Response pair as an element of a training set.

4.1 Nearest Neighbor graph-based classification

To identify an argument in a text, once the CDT is built, one needs to compute its similarity with CDTs for the positive class and verify that this similarity is lower than the similarities of this CDT with each element of the negative class.

Similarity between CDT is defined by means of maximal common sub-CDTs. Since we describe CDTs by means of labeled graphs, first we consider formal definitions of labeled graphs and of the domination relation on them (see, e.g., Ganter & Kuznetsov 2001).

Let's have an ordered set G of CDTs (V, E) with vertex- and edge-labels from the sets (Λ_v, \preceq) and (Λ_e, \preceq) . A labeled CDT Γ from G is a pair of pairs of the form $((V, l), (E, b))$, where V is a set of vertices, E is a set of edges, $l: V \rightarrow \Lambda_v$ is a function assigning labels to vertices, and $b: E \rightarrow \Lambda_e$ is a function assigning labels to edges. We do not distinguish isomorphic trees with identical labeling.

The order is defined as follows: For two CDTs $\Gamma_1 := ((V_1, l_1), (E_1, b_1))$ and $\Gamma_2 := ((V_2, l_2), (E_2, b_2))$ from G we say that Γ_1 **dominates** Γ_2 or $\Gamma_2 \leq \Gamma_1$ (or Γ_2 is a **sub-CDT** of Γ_1) if there exists a one-to-one mapping $\varphi: V_2 \rightarrow V_1$ such that it

- ∞ respects edges: $(v, w) \in E_2 \Rightarrow (\varphi(v), \varphi(w)) \in E_1$,
- ∞ fits under labels: $l_2(v) \preceq l_1(\varphi(v))$, $(v, w) \in E_2 \Rightarrow b_2(v, w) \preceq b_1(\varphi(v), \varphi(w))$.

This definition takes into account the calculation of similarity (“weakening”) of labels of matched vertices when passing from the “larger” CDT G_1 to “smaller” CDT G_2 .

Now, the similarity CDT Z of a pair of CDTs X and Y , denoted by $X \wedge Y = Z$, is the set of all inclusion-maximal common sub-CDTs of X and Y , each of them satisfying the following additional conditions:

- ∞ To be matched, two vertices from CDTs X and Y must denote the same RST relation;
- ∞ Each common sub-CDT from Z contains at least one communicative action with the same VerbNet signature as in X and Y .

This definition is easily extended to finding generalizations of several graphs (e.g., see Ganter and Kuznetsov 2001; Kuznetsov 1999). The subsumption order ∞ on pairs of graph sets X and Y is naturally defined as $X \infty Y := X * Y = X$.

An example of maximal common sub-CDT for CDTs Fig. 2a and 3b is shown in Fig. 6. Notice that the tree is inverted and the labels of arcs are generalized: Communicative action *cite()* is generalized with communicative action *say()*.

The first (agent) argument of the former CA *committee* is generalized with the first argument of the latter CA *Dutch*. The same operation is applied to the second arguments for this pair of CAs: *investigator* \wedge *evidence*. Notice that because the arguments of rhetorical relations are different in a general case, the leaf nodes on the bottom of the resultant Discourse Tree are empty.

We define the condition such that CDT U belongs to a positive class:

- 1) U is similar to (has a nonempty common sub-CDT) with a positive example R^+ .
- 2) For any negative example R^- , if U is similar to R^- (i.e., $U * R^- \neq \emptyset$) then $U * R^- \propto U * R^+$.

This condition introduces the measure of similarity and says that to be assigned to a class, the similarity between the unknown CDT U and the closest CDT from the positive class should be higher than the similarity between U and each negative example.

Condition 2 implies that there is a positive example R^+ such that for no R^- one has $U * R^+ \propto R^-$, i.e., there is no counterexample to this generalization of positive examples.

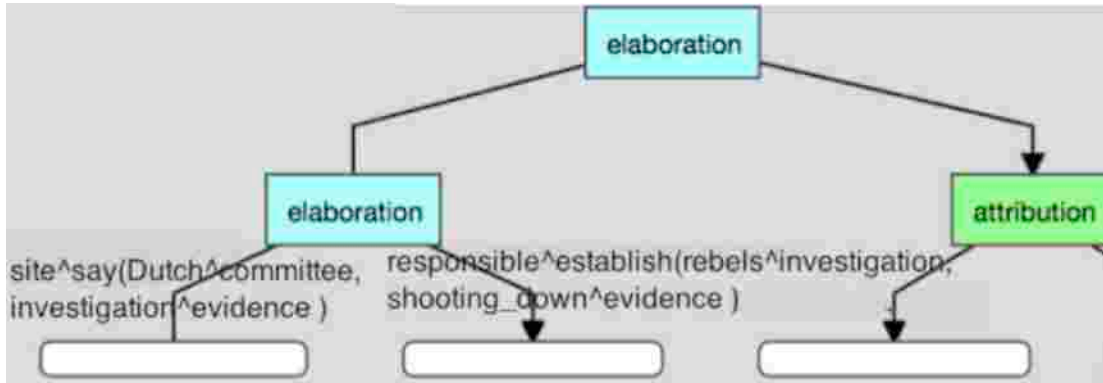


Fig. 6: Maximal common sub-CDT for two CDTs Figs. 2a and 2b

4.2 Thicket Kernel learning for CDT

Tree Kernel learning for strings, parse trees and parse thickets is a well-established research area these days. The parse tree kernel counts the number of common sub-trees as the discourse similarity measure between two instances. Tree kernels have been defined for DT by (Joty and Moschitti et al., 2014). (Wang et al., 2013) used a special form of tree kernels for discourse relation recognition. In this study we define the *thicket kernel* for CDT, augmenting DT kernels with the information on communicative actions.

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

$V(T) = (\# \text{ of subtrees of type } 1, \dots, \# \text{ of subtrees of type } I, \dots, \# \text{ of subtrees of type } n)$. This results in a very high dimensionality since the number of different sub-trees is exponential in its size. Thus, it is computationally infeasible to directly use the feature vector $\emptyset(T)$. To solve the computational issue, a tree kernel function is introduced to calculate the dot product between the

above high dimensional vectors efficiently. Given two tree segments CDT_1 and CDT_2 , the tree kernel function is defined:

$$K(CDT_1, CDT_2) = \langle V(CDT_1), V(CDT_2) \rangle = \sum_i V(CDT_1)[i], V(CDT_2)[i] = \sum_{n_1} \sum_{n_2} \sum_i I_i(n_1) * I_i(n_2)$$

where:

$n_1 \in N_1, n_2 \in N_2$ where N_1 and N_2 are the sets of all nodes in CDT_1 and CDT_2 , respectively;

$I_i(n)$ is the indicator function.

$I_i(n) = \{1 \text{ iff a subtree of type } i \text{ occurs with root at node; } 0 \text{ otherwise}\}.$

$K(CDT_1, CDT_2)$ is an instance of convolution kernels over tree structures (Collins and Duffy, 2002) and can be computed by recursive definitions:

$$\Delta(n_1, n_2) = \sum_i I_i(n_1) * I_i(n_2); (1)$$

$\Delta(n_1, n_2) = 0$ if n_1 and n_2 are assigned the same POS tag or their children are different subtrees. (2)

Otherwise, if both n_1 and n_2 are POS tags (are pre-terminal nodes) then $\Delta(n_1, n_2) = 1 \times \lambda$; (3)

Otherwise, $\Delta(n_1, n_2) = \lambda \prod_{j=1}^{nc(n_1)} (1 + \Delta(\text{ch}(n_1, j), \text{ch}(n_2, j)))$ (4)

where $\text{ch}(n, j)$ is the j^{th} child of node n , $nc(n_1)$ is the number of the children of n_1 , and λ ($0 < \lambda < 1$) is the decay factor in order to make the kernel value less variable with respect to the sub-tree sizes. In addition, the recursive rule (3) holds because given two nodes with the same children, one can construct common sub-trees using these children and common sub-trees of further offsprings. The parse tree kernel counts the number of common sub-trees as the syntactic similarity measure between two instances.

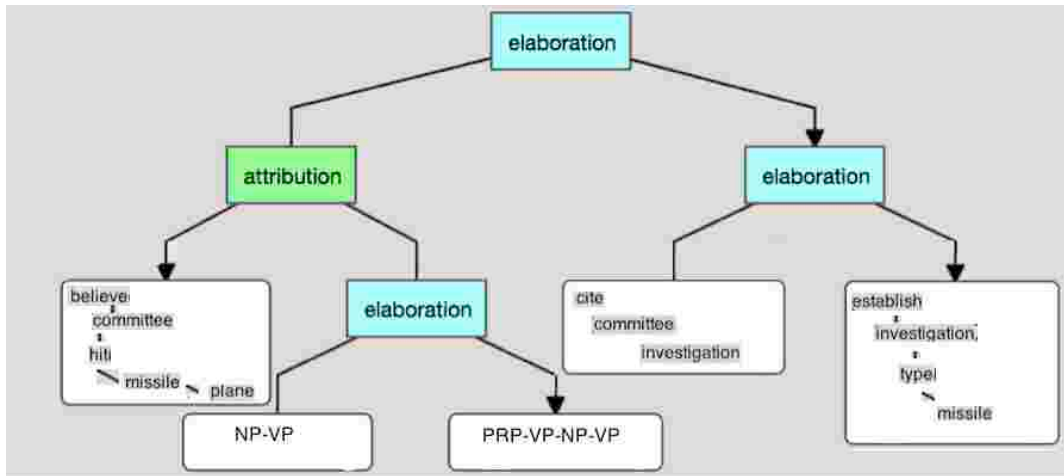
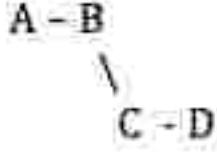


Fig.7: A tree in the kernel learning format for CDT Fig 2a.

A CDT representation for kernel learning is shown in Fig. 7. The terms for Communicative Actions as labels are converted into trees which are added to respective nodes for RST relations.

For texts for EDUs as labels for terminal nodes only the phrase structure is retained: we label the terminal nodes with the sequence of phrase types instead of parse tree fragments.

If there is a rhetorical relation arc from a node X to a terminal EDU node Y with label $A(B, C(D))$, then we append the subtree in Fig. 7a to X .



4.3 Implementation of the Rhetorical Agreement classifier

1. Define positive and negative classes of RR pairs:
 - a) Form the positive class from the rhetorically correct RR pairs
 - b) Form the negative class from the relevant but rhetorically foreign RR pairs
2. For each RR pair:
 - a) Parse each sentence
 - b) Obtain VerbNet structure for verbs
 - c) Obtain coreferences
 - d) Obtain entity - entity and entity – sub-entity links
 - e) Build parse thicket pair for PT_{RR}
 - f) Apply discourse parsing to obtain discourse tree pair DT_{RR} for RR pair
 - g) Align EDUs of DT_{RR} with PT_{RR}
 - h) Merge aligned EDUs of DT_{RR} with PT_{RR}
 - i) Obtain DT_{RR} with VerbNet signatures for CAs
 - j) Obtain parse thicket with enriched RST relations
 - k) Build representation for Thicket Kernel learning
 - l) Build representation for Nearest Neighbor learning
3. Apply Thicket Kernel learning
4. Apply Nearest Neighbor learning

Fig. 8a: Implementation of the Rhetorical Agreement classifier.

Fig. 8a shows the architecture of the Rhetorical Agreement Classifier. The training dataset stores the Request-Response pairs for positive and negative cases (on the top). Each sentence of each request and response is parsed and then combined into a parse thicket. For each word, we obtain POS, entity type, and for verbs we obtain VerbNet entries including roles and frames. Once a verb is identified, we use JVerbNet framework to attach the verb-related information to the node of the parse tree. An algorithm of populating the roles with words from respective role classes is straightforward. Logical form expressions which can potentially be attached to the nodes of discourse trees are also obtained from VerbNet.

VerbNet information is valuable for both learning setting: for nearest-neighbor, it helps to express similarity in a richer way when matching verbs are different. Without VerbNet information, the result of generalization is an empty lemma and POS=verb. With VerbNet, all common attributes shared by the different verbs being generalized are retained. For thicket kernel learning, VerbNet information helps to obtain a richer set of subgraphs with node labels for VerbNet attributes in the similar case where matching verbs are different. The reader is recommended to consult (Galitsky 2012) for further details of how phrases with VerbNet attributes are generalized.

To build a parse thicket, we form a graph from all parse tree nodes and add arcs for coreference, rhetorical and entity-entity relations. Discourse trees are initially built as a result of rhetorical parsing (Surdeanu et al., 2015) and then VerbNet labels are obtained as a result of their merge with parse thickets. We did not use our own training set for rhetorical parsing and used the trained parser instead. Notice that since parse thicket building is based on Stanford NLP and rhetorical parsing is also based on Stanford NLP, integration of these systems is not difficult.

To perform nearest neighbor learning we represent paragraphs as graphs and compute their maximal common subgraphs in terms of common phrases. To obtain these common phrases we used the generalization operation which is applied at the level of words, phrases, sentences and paragraphs (Galitsky et al., 2012).

Some of the components of the Rhetorical Agreement Classifier are implemented in the current study, some were implemented in our previous studies and a number of open source components developed by others were employed (please see Appendix). For the first two cases, we show the Java packages implementing functionality of a given component as a part of GitHub project <https://github.com/bgalitsky/relevance-based-on-parse-trees>.

The least reliable component of the Rhetorical Agreement Classifier is the rhetorical parser. Although splitting into EDUs works reasonably well, assignment of RST relation is noisy and in some domains its accuracy can be as low as 50% (personal communications from some users of discourse parsers). However, when the RST relation label is random, it does not significantly drop the performance of our classification system since a random discourse tree will be less similar to elements of positive or negative training set, and most likely will not participate in positive or negative decisions. To overcome the noisy input problem, more extensive training datasets are required so that the number of reliable, plausible discourse tree is high enough to cover cases to be classified. As long as this number is high enough, a contribution of noisy, improperly built discourse trees is low.

5 Evaluation

5.1 Evaluation domains

To evaluate rhetorical agreement in texts of various styles we form RR-pairs from various sources, applying distinct mechanisms to form positive and negative datasets. Table 1 shows the sources of evaluation and characterizes them in terms of the volume, average lengths of positive and negative sets in terms of sentences and words, and also the average numbers of rhetorical relations in respective discourse trees.

Our *first domain* is Yahoo! Answer set of question-answer pairs with broad topics (Chang et al., 2008). This dataset includes 20 top-level categories of Yahoo! Answer website. Out of the set of 4.4 million user questions we selected 20000, which included more than two sentences. Answers for most 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 one member of the negative training set. To derive the negative set, we either randomly select an answer to a different but somewhat related question, or formed a query from the question and obtained an answer from web search results.

Source	Yahoo! Answers			Conversation on Social Networks			Customer complaints			Interviews by journalists		
# of data items (<i>total / positive / negative</i>)	40k	20k	20k	2035	1232	803	670	415	255	3740	1870	1870
Average length of data items (in sentences)	2.8	3.1	2.5	3.9	4.0	3.9	5.2	4.7	5.5	2.9	4.1	1.8
Average length of data items (in words)	68	72	65	82	85	79	91	84	93	63	71	48
Average number of rhetorical relations	10.3	11.1	9.7	12.7	13.4	11.9	13.4	12.0	14.2	9.5	10.0	8.3
Average number of essential rhetorical relations	3.2	3.2	3.1	3.4	3.7	3.3	3.5	3.3	3.6	3.1	3.3	3.0

Table 1: Data sources

Our *second dataset* includes social media. We extracted Request-Response pairs mainly from postings on Facebook. We also used a smaller portion of LinkedIn.com and English vk.com conversations related to employment. In the social domains the standards of writing are fairly low. The cohesiveness of text is very limited and the logical structure and relevance frequently absent. The authors formed the training sets from their own accounts and also public Facebook accounts available via API over a number of years (at the time of writing, Facebook API for getting messages is unavailable). In addition, we used 860 email threads from the Enron dataset (Cohen 2016). Also, we collected the data of manual responses to postings of an agent which automatically generates posts on behalf of human users-hosts (Galitsky et al., 2014). We formed 2035 RR pairs from the various social network sources where request and response include 3+ grammatically correct sentences.

The *third domain* is customer complaints. In a typical complaint a dissatisfied customer describes his problems with products and service as well as the process for how he attempted to communicate these problems with the company and how they responded. Complaints are frequently written in a biased way, exaggerating product faults and presenting the actions of opponents as unfair and inappropriate. At the same time, the complainants try to write complaints in a convincing, coherent and logically consistent way (Galitsky et al., 2014, Github-DeceptionDataset 2017); therefore complaints serve as a domain with high agreement between requests and response. We split each complaint into two parts:

- 1) A complainant describing how he communicated an issue;
- 2) This complainant describing how the company responded.

For the purpose of assessing agreement between user complaint and company response (according to how this user describes it) we collected 670 complaints from planetfeedback.com over 10 years.

A typical complaint is a report of a failure of a product or service, followed by a narrative on the customer's attempts to resolve the issue. These complaints include both a description of the product or service failure and a description of the resulting interaction process (negotiation, conflict, etc.) between the customer and the company representatives. Since it is almost impossible to verify the actual occurrence of such failures, company representatives must judge the adequacy of a complaint on the basis of the communicative actions. A complaint narrative usually describes a conflict between an unsatisfied customer and customer support representatives, in which communicated claims need to be rationally justifiable by sound arguments. In contrast with the almost unlimited number of possible details regarding product failures, the emerging argumentative dialogues between customer and company can be subject to a systematic computational study (Galitsky et al., 2009).

The fourth domain is interviews by journalists. Usually, the way interviews are written by professional journalists is such that the match between questions and answers is very high. We collected 1200 contributions of professional and citizen journalists from such sources as allvoices.com, huffingtonpost.com and others. Over four years from 2011 to 2013, about 27 000 interviews by citizen journalists were submitted to AllVoices.com. These interviews contain an extended question by a journalist, providing some background and own journalist opinion along with the question or a statement. It is followed by an answer of the person being interviewed, written by the journalist. Also, some of journalists' interviews include the comments inserted by other website users, usually not journalists. Querying the database of articles and comments, we selected 1870 triples of paragraphs containing:

- 1) Original question and background;
- 2) Answer by the person being interviewed;
- 3) Comment by a user other than journalist.

We form the positive dataset from 1 and 2 and negative dataset with 1) and 3). Obviously 1) and 3) share the same topic but usually not in a good rhetorical agreement (as, for example, 1)+2) vs. 3) would be).

To facilitate data collection, we designed a crawler which searched a specific set of sites, downloaded web pages, extracted candidate text and verified that it adhered to a question-or-request vs. response format. Then the respective pair of text is formed. The search is implemented via Microsoft Cognitive Services Search API in the Web, News & Blogs domains.

Each data source in Table 1 is characterized by the following parameters with respective rows:

1. The number of data items (paragraphs) in the training set. We show the number of *total*, *positive* and *negative* cases;
2. Average length of data items (in sentences). We also show the number of *total*, *positive* and *negative* cases;
3. Average length in words;
4. Average number of rhetorical relations (measured as DT edges);
5. Average number of essential rhetorical relations (other than *elaboration* and *joint*)

5.2 Recognizing valid and invalid R-R pairs

In this section, we evaluate the accuracies of rhetorical agreement-based classification. We first outline the baseline methods for rhetorical agreement assessment, present the results for nearest

neighbor learning based on computing maximal common sub-graphs (Table 2a), and then proceed to the evaluation of the statistical, kernel-based methods (Table 2b). Each row represents a specific method.

The first baseline approach we selected in an attempt to detect proper rhetorical agreement was based on counting rhetorical relations. The premise here is that if a request and response have similar rhetorical relations, then agreement should be high. Conversely, if the rhetorical relations were different for this request and response, then we would expect the rhetorical agreement to be low. Hence, the linear regression-based classifier counts the different rhetorical relations. This premise turned out to provide too coarse of an approximation of the rhetorical agreement phenomenon: it was false in most cases, and the recognition accuracy was close to that of a random classifier.

The second baseline approach was based on a naïve hypothesis that common keywords might be correlated with rhetorical agreement. When computing the common keywords, we removed stop words as per the default search application (Lucene library). This hypothesis turned out not to be the case: the common keywords shared by Req and Resp were weakly correlated with rhetorical agreement.

Source / Evaluation setting	Yahoo! Answers			Conversation on Social Networks			Customer complaints			Interviews by journalists		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Counts of types of rhetorical relations of Req and Resp	55.2	52.9	54.03	51.5	52.4	51.95	54.2	53.9	54.05	53.0	55.5	54.23
Bag-of-words	54.7	55.2	54.94	50.8	51.9	51.90	55.0	52.7	53.83	51.6	54.3	52.92
Entity-based alignment of DTs of Req and Resp	63.1	57.8	60.33	51.6	58.3	54.70	48.6	57.0	52.45	49.2	57.9	53.21
Maximal common sub-DT for Req and Resp	67.3	64.1	65.66	70.2	61.2	65.40	54.6	60.0	57.16	80.2	69.8	74.61
Maximal common sub-CDT for Req and Resp	68.1	67.2	67.65	68.0	63.8	65.83	58.4	62.8	60.48	77.6	67.6	72.26

Table 2a: Evaluation results for baseline approaches and maximal common subgraph-based learning.

The third baseline approach relied on a named-entity-based alignment of the DTs of Req and Resp. The intuition behind this approach was that if the RR DTs are well-aligned in terms of these entities, the rhetorical agreement should be high. An abstract tree alignment problem extensively studied in bioinformatics is NP hard even when the set of labels is limited (Varon & Wheeler 2012). Therefore, we reduced it to a sequence alignment problem for fairly short sequences of entities; thus, no optimization was required. Although this approach resulted in an improvement over the previous one by a few percentile points, one might observe that matching entities between Req and Resp are weakly correlated with rhetorical agreement.

The fourth and fifth approaches were based on computing the maximal common sub-graph as a way to measure the similarity between trees, as presented in Section 4.1. For the richer set of labels provided by CDT versus DT, we obtained smaller common sub-trees but with a higher number of labels. One can observe that the rhetorical agreement classifier based on CDT provided a better performance than the one based on DT, exceeding the entity alignment case by a few percentiles.

Source / Evaluation setting	Yahoo! Answers			Conversation on Social Networks			Customer complaints			Interviews by journalists		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
SVM TK for Parse Trees of individual sentences	66.1	63.8	64.93	69.3	64.4	66.80	46.7	61.9	53.27	78.7	66.8	72.24
SVM TK for RST and CA (full parse thicket)	75.8	74.2	74.99	72.7	77.7	75.11	63.5	74.9	68.74	75.7	84.5	79.83
SVM TK for RR-DT	76.5	77	76.75	74.4	71.8	73.07	64.2	69.4	66.69	82.5	69.4	75.40
SVM TK for RR-CDT	80.3	78.3	79.29	78.6	82.1	80.34	59.5	79.9	68.22	82.7	80.9	81.78
SVM TK for RR-CDT + sentiment + argumentation features	78.3	76.9	77.59	67.5	69.3	68.38	55.8	65.9	60.44	76.5	74.0	75.21

Table 2b: Evaluation results for thicket learning family of approaches.

We present the classification results for the SVM TK family of approaches (Section 4.2) in Table 2b. From top to bottom, we extend the sources of linguistic information employed by SVM:

- 1) Parse trees for sentences only;
- 2) Parse trees are connected into parse thickets, so that rhetorical relations, communicative actions, coreference links and entity-entity links are leveraged;
- 3) Regular discourse trees;
- 4) Communicative discourse trees;
- 5) Communicative discourse trees with sentiment and argumentation features added.

One can see that the highest accuracy was achieved in the domains of *journalism and community answers* and the lowest in the domains of *customer complaints* and *social networks*. We can conclude that the higher the accuracy achieved by having the method fixed is, the higher the level of agreement between *Req* and *Resp* is. As the responder’s competence becomes higher, the rhetorical agreement increases as well.

The best representative of the deterministic family of approaches (the bottom row of Table 2a) performed approximately 9% below the best of SVM TK (the fourth row of Table 2b). This observation indicates that the similarity between *Req* and *Resp* is substantially less important than certain structures of RR pairs indicative of an RR agreement. This result means that the agreement between *Req* and *Resp* cannot be assessed on an individual basis: if we demand that $DT(Req)$ be very similar to $DT(Resp)$, we will obtain decent precision but extremely low recall.

Proceeding from DT to CDT for SVM TK helps by only 1–2%, because communicative actions play no major role in either composing a request or forming a response.

For the statistical family of approaches (Table 2b), the richest source of discourse data (SVM TK for RR-DT) provided the highest classification accuracy—almost the same as under the RR similarity-based classification. Although SVM TK for RST and CA (full parse trees) included more linguistic features some part of it (most likely syntactic) was redundant and yielded poorer results for the limited training set. Using additional features as input to TK, such as sentiment and argumentation, did not help either. Most likely, these features were derived from RR-CDT features and did not contribute to classification accuracy on their own.

To draw a comparison with the accuracy of rhetoric parsers, we state that employing the TK family of approaches based on CDT provided accuracy comparable to that achieved by classifying DT as correct or incorrect by the rhetorical parsing tasks on which state-of-the-art systems have competed over the last few years approached an accuracy of over 80%.

Direct analytical approaches in the deterministic family performed rather weakly, which means that a higher number and a more complicated structure of features is required. Simply counting and considering types of rhetorical relations was insufficient to judge how the RR agreed with each other. Even when two RR pairs have the same types and counts of rhetorical relations and communicative actions, they could still belong to opposite RR agreement classes in most cases.

Nearest-pair neighbor learning for CDT achieved lower accuracy than did SVM TK for CDT, but the former yielded interesting examples of sub-trees typical of argumentation and ones that are shared among the Q/A pairs of the factoid type. The number of the former groups of CDT sub-trees was naturally significantly higher. Unfortunately, the SVM TK approach did not help in explaining exactly how the RR agreement problem was solved. It only provided final scoring and class labels. It is possible but uncommon to express a logical argument in a response without communicative actions (this observation was backed up by our data).

Manual analysis of the false negative RR pairs showed that some cases of rhetorical agreement were hard to cover by the mapping of respective CDTs. A number of responses that could be viewed as *sarcastic* did not follow rhetorical structure but were nevertheless considered by the readers as cohesive. Many cases of *official* answers, answers in *legalese* or other domain-specific professional language were wrongly classified as being in bad agreement. This was not only because of noisier CDT construction but also due to the limitation of the CDT model in general. We believe an additional model for RR pair agreement is required, one that goes beyond CDT mapping and discourse level in general. Creating such a model could be a topic of future studies.

Most false-positive RR pairs were obtained when the CDT of the request was structurally similar to the CDT of the response but when the rhetorical agreement was nevertheless low. These were mostly cases in which the RR pair to be classified was similar to a given RR pair in the positive training dataset but the rhetorical structure was not suitable for a given domain. The next step in improving a rhetorical agreement classifier would be to have a training set specific to a broad domain rather than a domain-independent training set formed for a given text genre, as we did in this study. This possibility could potentially be explored in future research.

5.3 CDT Construction Task

In this section, we evaluate how well CDTs were constructed irrespective of how they were used and learned. As mentioned earlier, Although splitting text into EDUs works reasonably well, assignment of the RST relation is fairly noisy. However, when the RST relation label is random, it does not significantly drop the performance of our argumentation detection system. To overcome the noisy input problem, more extensive training datasets are required to make the number of reliable, plausible discourse trees high enough to cover the cases to be classified.

When this number is sufficiently high, the contribution of noisy, improperly built discourse trees is low.

There is a certain systematic deviation from the correct, intuitive discourse trees obtained by discourse parsers. In this section, we evaluate whether there is a correlation between the deviation in CDTs and some specificities of our training sets. We consider the possibility that the CDT deviations for Q/A pairs with high rhetorical agreement is stronger than the ones with low rhetorical agreement.

For each source, we calculate the number of significantly deviated CDTs. For this assessment, we consider a CDT as deviated when more than 20% of rhetorical relations were determined improperly. We did not differentiate between the specific RST relations associated with high rhetorical agreement. The CDT distortion evaluation dataset was significantly smaller than the detection dataset, because substantial manual effort was required, and the task could not be submitted to Amazon Mechanical Turk workers.

As Table 3 shows, there was no obvious correlation between the recognition classes and the rate of CDT distortion (less than 3%). Hence, we conclude that the training set of noisy CDTs can be adequately evaluated with respect to the detection of high and low rhetorical agreement.

Source	Positive training set size	Negative training set size	Significantly deviating DTs for Positive training set, %	Significantly deviating DTs for Negative training set, %
Yahoo! Answers	50	50	18.6±3.43	21.3±2.34
Conversation on Social Networks	50	50	16.0±4.92	19.8±5.40
Customer complaints	40	40	21.3±4.62	18.8±4.36
Interviews	40	40	17.6±3.43	18.5±4.72

Table 3: Does deviation in CDT construction depend on the domain?

We also assessed the agreement between the two rhetorical parsers available from Surdeanu et al. (2015) and Joty et al. (2013). In approximately 60% of the cases, the resultant DTs did not match, but in less than 15% of cases, this deviation affected the rhetorical agreement decision, as a small subset of our evaluation set shows. We did not collect evidence on which parser performed better in our domains; we used only the Surdeanu et al.’s (2015) parser in this study due to its ease of integration and its lightweight implementation.

6 Related Work

Although discourse analysis has a limited number of applications in question-answering and summarization and generation of text, we have not found applications of automatically constructed discourse trees. We discuss research related to applications of discourse analysis in

two areas: dialogue management and dialogue games. These areas can potentially be applied to the same problems as those for which the current proposal is intended. Research in these areas includes both logic-based approaches as well as analytical and machine learning-based approaches.

6.1 Managing dialogues and question answering

If a question and answer are logically connected, their rhetorical structure agreement becomes less important.

(De Boni 2007) proposed a method of determining the appropriateness of an answer to a question through a proof of logical relevance rather than a logical proof of truth. We define logical *relevance* as the idea that answers should not be considered as absolutely true or false in relation to a question, but should be considered true more flexibly in a sliding scale of aptness. Then it becomes possible to reason rigorously about the appropriateness of an answer even in cases where the sources of answers are incomplete or inconsistent or contain errors. The authors show how logical relevance can be implemented through the use of measured simplification, a form of constraint relaxation, in order to seek a logical proof that an answer is in fact an answer to a particular question.

Our model of CDT attempts to combine general rhetorical and speech act information in a single structure. While speech acts provide a useful characterization of one kind of pragmatic force, more recent work, especially in building dialogue systems, has significantly expanded this core notion, modeling more kinds of conversational functions that an utterance can play. The resulting enriched acts are called *dialogue acts* (Jurafsky and Martin, 2000). In their multi-level approach to conversation acts (Traum and Hinkelman 1992) distinguish four levels of dialogue acts necessary to assure both coherence and content of conversation. The four levels of conversation acts are: turn-taking acts, grounding acts, core speech acts, and argumentation acts.

Research on the logical and philosophical foundations of Q/A has been conducted over a few decades, having focused on limited domains and systems of rather small size and been found to be of limited use in industrial environments. The ideas of logical proof of “being an answer to” developed in linguistics and mathematical logic have been shown to have a limited applicability in actual systems. Most current applied research, which aims to produce working general-purpose (“open-domain”) systems, is based on a relatively simple architecture, combining Information Extraction and Retrieval, as was demonstrated by the systems presented at the standard evaluation framework given by the Text Retrieval Conference (TREC) Q/A track.

(Sperber and Wilson 1986) judged answer relevance depending on the amount of effort needed to “prove” that a particular answer is relevant to a question. This rule can be formulated via rhetorical terms as Relevance Measure: *the less hypothetical rhetorical relations are required to prove an answer matches the question, the more relevant that answer is*. The effort required could be measured in terms of amount of prior knowledge needed, inferences from the text or assumptions. In order to provide a more manageable measure we propose to simplify the problem by focusing on ways in which constraints, or rhetorical relations, may be removed from how the question is formulated. In other words, we measure how the question may be simplified in order to prove an answer. The resultant rule is formulated as follows: *The relevance of an answer is determined by how many rhetorical constraints must be removed from the question for the answer to be proven; the less rhetorical constraints must be removed, the more relevant the answer is*.

There is a very limited corpus of research on how discovering rhetorical relations might help in Q/A. (Santosh and Jahfar 2012) discuss the role of discourse structure in dealing with ‘why’ questions, that helps in identifying the relationship between sentences or paragraphs from a given

text or document. (Kontos et al., 2016) introduced a system which allowed an exploitation of rhetorical relations between a “basic ” text that proposes a model of a biomedical system and parts of the abstracts of papers that present experimental findings supporting this model.

Adjacency pairs is a popular term for what we call RR-pair in this paper. Adjacency pairs are defined as pairs of utterances that are adjacent, produced by different speakers, ordered as first part and second part, and typed—a particular type of first part requires a particular type of second part. Some of these constraints could be dropped to cover more cases of dependencies between utterances (Popescu-Belis 2005).

Adjacency pairs are relational by nature, but they could be reduced to labels (‘first part’, ‘second part’, ‘none’), possibly augmented with a pointer towards the other member of the pair. Frequently encountered observed kinds of adjacency pairs include the following ones: *request / offer / invite* → *accept / refuse*; *assess* → *agree / disagree*; *blame* → *denial / admission*; *question* → *answer*; *apology* → *downplay*; *thank* → *welcome*; *greeting* → *greeting* (Levinson 2000).

Rhetorical relations, similarly to adjacency pairs, are a relational concept, concerning relations between utterances, not utterances in isolation. It is however possible, given that an utterance is a satellite with respect to a nucleus in only one relation, to assign to the utterance the label of the relation. This poses strong demand for a deep analysis of dialogue structure. The number of rhetorical relations in RST ranges from the ‘dominates’ and ‘satisfaction-precedes’ classes used by (Grosz and Sidner 1986) to more than a hundred types. Coherence relations are an alternative way to express rhetorical structure in text (Scholman et al., 2016).

(Mitocariu et al., 2013) considers cases when two different tree structures of the same text can express the same discourse interpretation, or something very similar. The authors apply both RST and Veins Theory (Cristea et al., 1998), which uses binary trees augmented with nuclearity notation. In the current paper we attempt to cover these cases by learning, expecting different DTs for the same text to be covered by an extended training set.

There are many classes of NLP applications that are expected to leverage the informational structure of text. DT can be very useful in text summarization. Knowledge of salience of text segments, based on nucleus-satellite relations proposed by (Sparck-Jones 1995) and the structure of relation between segments should be taken into account to form exact and coherent summaries. One can generate the most informative summary by combining the most important segments of elaboration relations starting at the root node. DTs have been used for multi-document summaries (Radev 2000).

In the natural language generation problem, whose main difficulty is coherence, the informational structure of the text can be relied upon to organize the extracted fragments of text in a coherent way. A way to measure text coherence can be used in automated evaluation of essays. Since a DT can capture text coherence, then yielding discourse structures of essays can be used to assess the writing style and quality of essays. (Burstein et al., 2002) described a semi-automatic way for essay assessment that evaluated text coherence.

The neural network language model proposed in (Bengio et al., 2003) uses the concatenation of several preceding word vectors to form the input of a neural network, and tries to predict the next word. The outcome is that after the model is trained, the word vectors are mapped into a vector space such that Distributed Representations of Sentences and Documents semantically similar words have similar vector representations. This kind of model can potentially operate on discourse relations, but it is hard to supply as rich linguistic information as we do for tree kernel learning. There is a corpus of research that extends word2vec models to go beyond word level to achieve phrase-level or sentence-level representations (Mikolov et al., 2015). For instance, a simple approach is using a weighted average of all the words in the document, (weighted averaging of word vectors), losing the word order similar to bag-of-words approaches. A more

sophisticated approach is combining the word vectors in the order given by a parse tree of a sentence, using matrix-vector operations (Socher et al., 2010). Using a parse tree to combine word vectors, has been shown to work only for sentences because it relies on parsing.

Given a DT for a text as a candidate answer to a compound query, (Galitsky et al., 2015) proposed a rule system for valid and invalid occurrence of the query keywords in this DT. To be a valid answer to a query, its keywords need to occur in a chain of elementary discourse units of this answer so that these units are fully ordered and connected by nucleus – satellite relations. An answer might be invalid if the queries’ keywords occur in the answer’s satellite discourse units only.

Classifying user intent in horizontal web searches is a well-known difficult problem. Jansen et al. (2007) proposed a classification algorithm to determine user intent underlying Web search engine queries that considers three classes: informational, navigational, and transactional. The results showed that more than 80% of web queries are informational in nature, while approximately 10% each are navigational or transactional. Kathuria et al. (2010) solved the same problem using a k-means clustering approach based on a variety of query traits. The authors showed that 75% of web queries (clustered into eight classifications) are informational, while 12% each are navigational or transactional. Their results also show that web queries fall into eight clusters, six are primarily informational and two are primarily transactional or navigational.

As for a chatbot, one of its essential capabilities is to discriminate between a request to commit a transaction and a question to obtain some information (Galitsky et al., 2017). This capability is expected to be domain-independent: in any domain, a user may either request that the system do something or provide a recommendation. This functionality should also be context-independent: a user may switch from information access to a request to do something and back to information access, although this should be discouraged. Even human customer support agents prefer a user to first receive information, then make a decision, and finally, request an action. Lewandowski (2016) confirmed that search engine performance on navigational queries is of great importance, because users can clearly identify queries that have returned correct results. As such, performance on navigational query types may contribute to explaining user satisfaction with search engines.

The performance of chat bots and search engines strongly depends on their ability to capture user intent. Vogel et al. (2005) addressed the issue of mapping a search engine query to certain nodes of a subject taxonomy that expresses a possible query. An architecture of a user intent classification system uses a web directory to determine the query context by the query term frequencies.

6.2 Analytical approaches to RR Agreement

In this paper, we formulated the problem of RR agreement and approached it via machine learning. However, there are several approaches other than learning that tackle the relationships between request and response from different perspectives. Not all features of these perspectives are covered by our learning framework, which is designed to automatically extract available features from text. Moreover, a statistical or reinforcement learning framework does not reveal which features exactly are leveraged (Rieser and Lemon 2011). Therefore, it is worth mentioning explicit models of relationships between requests and responses.

In an arbitrary conversation, a question is typically followed by an answer, or some explicit statement of an inability or refusal to answer. The following model explains the intentional thread of a conversation as: From the yielding of a question by Agent *B*, Agent *A* recognizes Agent *B*’s goal to find out the answer, and it adopts a goal to tell *B* the answer in order to be co-operative. *A* then plans to achieve the goal, thereby generating the answer. This provides an elegant account in the simple case, but requires a strong assumption of co-cooperativeness. Agent

A must adopt agent *B*'s goals as her own. As a result, it does not explain why *A* says anything when she does not know the answer or when she is not ready to accept *B*'s goals.

An intentional analysis at the level of textual discourse was introduced by (Litman and Allen 1987). The authors assumed a set of typical multi-agent actions. Other authors attempted to simulate these forms of multiagent behavior via social intentional models such as *Joint intentions* (Cohen and Levesque 1991) or *Shared Plans* (Grosz and Sidner 1990). Although these approaches help explain certain discourse features of multiagent interaction, they still need to shed a light on how dialogue coherence is achieved.

Let us imagine a stranger approaching a person and asking, "Do you have spare coins?" It is unlikely that there is a joint intention or shared plan, as they have never met before. From a purely strategic point of view, the agent may have no interest in whether the stranger's goals are met. Yet, typically agents will still respond in such situations. Hence an account of Q/A must go beyond recognition of speaker intentions. Questions do more than just provide evidence of a speaker's goals, and something more than adoption of the goals of an interlocutor is involved in formulating a response to a question.

An interesting model is described by (Airenti et al., 1993), which separates out the conversational games from the task-related games in a way similar way to (Litman and Allen 1987). Because of this separation, they do not have to assume co-operation on the tasks each agent is performing, but still require recognition of intention and co-operation at the conversational level. It is left unexplained what goals motivate conversational co-operation.

6.3 Rhetorical relations and argumentation

Frequently, the main means of linking questions and answers is logical argumentation. There is an obvious connection between RST and argumentation relations which we tried to learn in this study. There are four types of rhetorical relations correlated with logical argumentation: the directed relations support, attack, detail, and the undirected sequence relation (Lippi and Torroni 2016). The support and attack relations are argumentative relations, which are known from related work (Peldszus and Stede, 2013), whereas the latter two correspond to discourse relations used in RST. The argumentation sequence relation corresponds to "Sequence" in RST, the argumentation *detail* relation roughly corresponds to "Background" and "Elaboration".

The argumentation *detail* relation is important because there are many cases in scientific publications, where some background information (for example the definition of a term) is important for understanding the overall argumentation. A support relation between an argument component *Resp* and another argument component *Req* indicates that *Resp* supports (reasons, proves) *Req*. Similarly, an attack relation between *Resp* and *Req* is annotated if *Resp* attacks (restricts, contradicts) *Req*. The detail relation is used, if *Resp* is a *detail* of *Req* and gives more information or defines something stated in *Req* without argumentative reasoning. Finally, we link two argument components (within *Req* or *Resp*) with the sequence relation, if they belong together and only make sense in combination, i.e., they form a multi-sentence argument component.

In our previous papers we observed that using SVM TK, one can differentiate between a broad range of text styles (Galitsky et al., 2015). In particular, based on rhetorical structure, it is possible to differentiate between documents without argumentation and ones with various forms of argumentation. This is also applicable to text with and without string sentiment, writing with strong legal focus, engineering / design document focus and a focus in finance. Each text style and genre has its inherent rhetorical structure that is leveraged and automatically learned. Since the correlation between text style and text vocabulary is rather low, traditional classification approaches which only take into account keyword statistics information could lack accuracy in

complex cases. We also performed text classification into rather abstract classes such as the belonging to language-object and metalanguage in literature domain and style-based document classification into proprietary design documents (Galitsky 2016). Evaluation of text integrity in the domain of valid versus invalid customer complains (those with argumentation flow, non-cohesive, indicating a bad mood of a complainant) shows the stronger contribution of rhetorical structure information in comparison with the sentiment profile information. Discourse structures obtained by the RST parser are sufficient to conduct the text integrity assessment, whereas sentiment profile-based approach shows much weaker results and also does not complement strongly the rhetorical structure ones. Handling dialogues via machine learning of communicative discourse trees allowed us to model a wide array of dialogue types of collaboration modes (Blaylock et al., 2003) and interaction types (planning, execution, and interleaved planning and execution).

7 Conclusion

An extensive corpus of studies has been devoted to RST parsers, but the research on how to leverage RST parsing results for practical NLP problems is limited to content generation, summarization and search (Jansen et al., 2014). DTs obtained by these parsers cannot be used directly in a rule-based manner to filter or construct texts. Therefore, learning is required to leverage the implicit properties of DTs. To our knowledge, this study is the only one that employs discourse trees and their extensions for general and open-domain question-answering.

Search engines and recommendation systems need to be capable of understanding and matching users' communicative intentions, reasoning with these intentions, building their own respective communication intentions and populating these intentions with actual language to be communicated to the user. Discourse trees on their own do not provide representations for these communicative intents. In this study, we introduced communicative discourse trees, built upon traditional discourse trees, which, on one hand, can currently be computed efficiently and, on the other hand, constitute a descriptive utterance-level model of a request-response pair.

Statistical computational learning approaches offer several key potential advantages over the manual rule-based hand-coding approach when developing search and recommendation systems:

- ∞ data-driven development cycle;
- ∞ provably optimal action policies;
- ∞ a more accurate model for response selection;
- ∞ possibilities for generalization to unseen states;
- ∞ reduced development and deployment costs for industry.

Comparing inductive learning results with kernel-based statistical learning while relying on the same information allowed us to perform more concise feature engineering than either approach would on their own. The task of comparing tree structures such as parse trees, discourse trees and parse thickets with respect to similarity is fairly important. Dot products of the vectors of features of the trees are adequate ways to implement similarity comparisons, but these vectors are multi-dimensional. Instead of representing complex structures such as parse trees and parse thickets with feature vectors, tree kernels are used, which allow for the computation of similarity over trees without explicitly computing the feature vectors of these tree structures. Kernel Methods and SVM in particular have been widely used in machine learning tasks and, therefore, are assumed to be the best approach to handle parse thickets and extended discourse trees.

Structural features can potentially be expressed in a deep learning framework; however, industrial deployment of such features in a major cloud infrastructure such as Oracle's would be difficult. The deep learning class of algorithm lacks feature explainability and feature engineering, which are the strong points of tree kernel learning, especially the nearest neighbor

framework. Real-time performance and a lack of large training datasets of discourse structures are additional factors favoring SVM and nearest-neighbor classes of learning algorithms.

RST parsers are mostly evaluated with respect to agreement with test sets annotated by humans rather than their expressiveness of the features of interest. In this work, we focused on interpretation of DTs and explored ways to represent them in a form indicative of an agreement or disagreement rather than the neutral enumeration of facts.

To provide a measure of agreement for how a given message in a dialogue is followed by a subsequent message, we used CDTs that include labels for communicative actions in the form of substituted VerbNet frames. We investigated the discourse features that were indicative of correct versus incorrect request-response and question-answer pairs. We used two learning frameworks to recognize correct pairs: deterministic, nearest-neighbor learning of CDTs as graphs and a tree kernel learning of CDTs, in which a feature space of all the CDT sub-trees was subject to SVM learning.

The positive training set was constructed from the correct pairs obtained from Yahoo Answers, social networks, corporate conversations (including Enron emails), customer complaints and interviews by journalists. The corresponding negative training set was created by attaching responses for different random requests and questions that included relevant keywords so that the relevance similarity between requests and responses was high. The evaluation showed that it is possible to recognize valid pairs in 68–79% of cases in the domains of weak request-response agreement and 80–82% of cases in the domains of strong agreement. These accuracy rates are essential to support automated conversations and are comparable to the benchmark task of classifying discourse trees as either valid or invalid as well as with factoid question-answering systems.

We believe that this study is the first one to leverage automatically built discourse trees for question-answering support. Previous studies have used specific customer discourse models and features that are hard to systematically collect, learn with explainability, reverse engineer and compare with each other. We conclude that learning rhetorical structures in the form of CDTs is a key source of data to support answering complex questions, chatbots and dialogue management.

The code used in this study is open source and available at: <https://github.com/bgalitsky/relevance-based-on-parse-trees>.

Acknowledgements

The author is grateful to his colleagues Dmitri Ilvovsky, Sergey Kuznetsov, Dina Pisarevskaya, Vishal Vishnoi, Stephen McRitchie, Gautam Singaraju, Kim Kanzaki, Mark Mathison for fruitful discussion, and to Barbara Di Eugenio, action editor, and anonymous reviewers for significant help in improving the manuscript.

References

- Laura Aina, Natalia Philippova, Valentin Vogelmann and Raquel Fernández (2017). Referring Expressions and Communicative Success in Task-oriented Dialogues. SemDial 2017 Saarbrücken Germany.
- Gabriella Airenti, Bara, Bruno G. and Colombetti, Marco (1993). Conversation and behavior games in the pragmatics of dialogue. *Cognitive Science* 17:197–256.
- James Allen, and C. Perrault (1980). Analyzing Intention in Utterances, *Artificial Intelligence*, 15(3):143–178,.
- Roy F. Baumeister and Bushman, B. J. (2010). *Social psychology and human nature: International Edition*. Belmont, USA: Wadsworth.

- Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. (2003). A neural probabilistic language model. *J. Mach. Learn. Res.* 3 (March 2003), 1137-1155.
- Nate Blaylock, James Allen and Ferguson, G. (2003). Managing Communicative Intentions with Collaborative Problem Solving. In *Current and New Directions in Discourse and Dialogue*, Springer Netherlands, Dordrecht, 63-84.
- David M. Blei, Ng, Andrew Y., and Jordan, Michael (2003). Latent Dirichlet Allocation. in Lafferty, John, ed. *Journal of Machine Learning Research*. 3 (4-5): pp. 993-1022. doi:10.1162/jmlr.2003.3.4-5.993.
- Jill C. Burstein, Lisa Braden-Harder, Martin S. Chodorow, Bruce A. Kaplan, Karen Kukich, Chi Lu, Donald A. Rock and Susanne Wolff (2002). System and method for computer-based automatic essay scoring. United States Patent 6,366,759: Educational Testing Service.
- Ming-Wei Chang, L. Ratinov, D. Roth and V. Srikumar (2008). Importance of Semantic Representation: Dataless Classification AAAI – 2008.
- Philip R. Cohen & Levesque, H. J. (1990). Intention is choice with commitment, *Artificial Intelligence*, 42: 213-261.
- William Cohen (2016). Enron Email Dataset . <https://www.cs.cmu.edu/~enron/> Last downloaded July 10, 2016.
- CrimeRussia (2016). <http://en.crimerussia.ru/corruption/shadow-chairman-of-the-investigative-committee>.
- Dan Cristea, Ide, N., & Romary, L. (1998). Veins theory: A model of global discourse cohesion and coherence. In C. Boitet & P. Whitelock (Eds.), *17th international conference on Computational linguistics* (Vol. 1 pp. 281-285). Montreal, Canada: Association for Computational Linguistics.
- Marco De Boni (2007). Using logical relevance for question answering, *Journal of Applied Logic*, Volume 5, Issue 1, March 2007, Pages 92-103.
- Vanessa Wei Feng and Hirst, G. (2011). Classifying Arguments by Scheme. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, Portland, OR, 987-996.
- Vanessa Wei Feng and Graeme Hirst (2014). A linear-time bottom-up discourse parser with constraints and post-editing. *ACL - 2014*, Baltimore, USA, June.
- Gov Gabbay and Garcez, A.S. Logical Modes of Attack in Argumentation Networks. *Stud Logica* (2009) 93: 199.
- Boris Galitsky, Ilvovsky, D. and Kuznetsov SO (2015). Rhetoric Map of an Answer to Compound Queries Knowledge Trail Inc. *ACL 2015*, 681-686.
- Boris Galitsky, Dmitri Ilvovsky, Nina Lebedeva and Daniel Usikov (2014) Improving Trust in Automation of Social Promotion. *AAAI Spring Symposium on The Intersection of Robust Intelligence and Trust in Autonomous Systems* Stanford CA.
- Boris Galitsky (2013). Content inversion for user searches and product recommendations systems and methods. US Patent 9336297.
- Boris Galitsky (2012). Machine learning of syntactic parse trees for search and classification of text. *Engineering Application of AI* . Volume 26, Issue 3, Pages 1072-1091
- Boris Galitsky (2016) Using extended tree kernels to recognize metalanguage in text. *Uncertainty Modeling*, in Kreinovich V., editor. Springer.
- Boris, Galitsky and Josep Lluís de la Rosa. (2011). Concept-based learning of human behavior for customer relationship management. *Special Issue on Information Engineering Applications Based on Lattices. Information Sciences*. Volume 181, Issue 10, 15 May 2011, pp 2016-2035.
- Boris Galitsky, Gabor Dobrocsi, Josep Lluís de la Rosa (2012). Inferring the semantic properties of sentences by mining syntactic parse trees. *Data & Knowledge Engineering*. Volume 81-82, November, 2012. Pages 21-45.
- Boris Galitsky, MP González, CI Chesñevar (2009). A novel approach for classifying customer complaints through graphs similarities in argumentative dialogue. *Decision Support Systems*, 46-3, 717-729.
- Boris Galitsky, Vishnoi, Vishal and Xu, Anfernee. (2017). Transaction Bot Discrimination between User's Question or Request. Oracle Provisional Patent Application 62/564,868.

- Github-DeceptionDataset (2017) <https://github.com/bgalitsky/relevance-based-on-parse-trees/blob/master/examples/ultimateDeception.xls>;
- Barbara J. Grosz and Candace Sidner (1986). Attention, Intention, and the Structure of Discourse, *Computational Linguistics*, 12(3):175–204.
- Barbara J. Grosz & Sidner, Candace L. (1986). Attentions, Intentions and the Structure of Discourse. *Computational Linguistics*, 12(3), 175– 204.
- Susan Haller, Susan McRoy, Alfred Kobsa (2013). *Computational Models of Mixed-Initiative Interaction*. Springer Science & Business Media, Nov 11, - Computers - 398 pages.
- Lee Heeyoung, Angel Chang, Yves Peirsman, Nathanael Chambers, Mihai Surdeanu and Dan Jurafsky. (2013). Deterministic coreference resolution based on entity-centric, precision-ranked rules. *Computational Linguistics* 39(4).
- Takuya Y. Hiraoka, Yamauchi, G. Neubig, S. Sakti, T. Toda and S. Nakamura (2013). Dialogue management for leading the conversation in persuasive dialogue systems, 2013 IEEE Workshop on Automatic Speech Recognition and Understanding, Olomouc, pp. 114-119.
- Hospice Houngho and Robert Mercer (2014). An automated method to build a corpus of rhetorically-classified sentences in biomedical texts. *Proceedings of the First Workshop on Argumentation Mining*, pages 19–23, Baltimore, Maryland USA, June 26, ACL.
- Mikel Iruskieta, Iria da Cunha and Maite Taboada (2015). A qualitative comparison method for rhetorical structures: identifying different discourse structures in multilingual corpora. *Lang Resources & Evaluation*, Volume 49, Issue 2, pp 263–309.
- Bernard J. Jansen, Danielle L. Booth, and Amanda Spink (2007). Determining the user intent of web search engine queries. In *Proceedings of the 16th international conference on World Wide Web (WWW '07)*. ACM, New York, NY, USA, 1149-1150.
- Peter Jansen, M. Surdeanu, and P. Clark (2014.) *Discourse Complements Lexical Semantics for Nonfactoid Answer Reranking*. In *Proceedings of the 52nd ACL*.
- Shafiq R. Joty, and A. Moschitti (2014). Discriminative Reranking of Discourse Parses Using Tree Kernels. *Proceedings of EMNLP*, 2049-2060.
- Shafiq R. Joty, Giuseppe Carenini, Raymond T. Ng (2016). CODRA: A Novel Discriminative Framework for Rhetorical Analysis. *Computational Linguistics* Volume 41, Number 3.
- Shafiq R. Joty, Giuseppe Carenini, Raymond T. Ng, and Yashar Mehdad. (2013). Combining intra-and multi- sentential rhetorical parsing for document-level dis- course analysis. In *ACL (1)*, pages 486–496.
- Daniel Jurafsky, James H. Martin (2000). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Upper Saddle River, NJ: Prentice Hall.
- Ashish Kathuria, Bernard J. Jansen, Carolyn Hafernik, Amanda Spink (2010). Classifying the user intent of web queries using k-means clustering, *Internet Research*, Vol. 20 Issue: 5, pp.563-581,
- Karin Kipper, Anna Korhonen, Neville Ryant and Martha Palmer. (2008). A large-scale classification of English verbs. *Language Resources and Evaluation Journal*, 42:21–40.
- John Kontos, Ioanna Malagardi, John Peros (2016). Question Answering and Rhetoric Analysis of Biomedical Texts in the AROMA System. Unpublished Manuscript. <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.379.5382> (last downloaded September 12, 2016).
- Stephen C. Levinson (2000). *Presumptive Meanings: The Theory of Generalized Conversational Implicature*. Cambridge, MA: The MIT Press.
- D. Lewandowski (2015). Evaluating the retrieval effectiveness of web search engines using a representative query sample. *J Assn Inf Sci Tec*, 66: 1763–1775.
- Z. Lin, Kan M. and Ng H. (2009). Recognizing Implicit Discourse Relations in the Penn Discourse Tree-bank. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing (EMNLP 2009)*, Singapore, August.

- Marco Lippi, Paolo Torrioni. (2016). Argumentation Mining: State of the Art and Emerging Trends. *ACM Trans. Internet Technol.* 16, 2, Article 10 (March 2016), 25 pages.
- Diancie Litman, James Allen (1987). A plan recognition model for subdialogues in conversation, *Cognitive Science*, 11: 163-200.
- William Mann, Matthiessen, C., Thompson, S (1992). *Rhetorical Structure Theory and Text Analysis. Discourse Description: Diverse linguistic analyses of a fund-raising text* / ed. by W. C. Mann and S. A. Thompson. – Amsterdam. –P. 39–78.
- 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.
- Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. (2014). The Stanford CoreNLP Natural Language Processing Toolkit. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pp. 55-60.
- Tomas Mikolov, I. Sutskever, K. Chen (2011). Distributed representations of words and phrases and their compositionality. in GS Corrado, J Dean. *Advances in Neural Information Processing Systems*, 3111-3119.
- Tomas Mikolov, K. Chen, G.S. Corrado; J. Dean (2015). Computing numeric representations of words in a high-dimensional space. US Patent 9,037,464, Google, Inc.
- Mitocariu, Elena, Daniel-Alexandru Anechitei, Dan Cristea, Comparing Discourse Tree Structures (2016) https://www.researchgate.net/publication/262331642_Comparing_Discourse_Tree_Structures [accessed May 15, 2016].
- Alessandro Moschitti, (2006). Efficient convolution kernels for dependency and constituent syntactic trees. in: *Proceedings of the 17th European Conference on Machine Learning*, Berlin, Germany.
- Alessandro Moschitti, S. Quarteroni (2011). Linguistic kernels for answer re-ranking in question answering systems, *Inf. Process. Manage.* 47 (6) 825–842.
- Myle Ott, Y. Choi, C. Cardie, and J.T. Hancock (2011). Finding Deceptive Opinion Spam by Any Stretch of the Imagination. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*.
- Myle Ott, C. Cardie, and J.T. Hancock (2013). Negative Deceptive Opinion Spam. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.
- Andreas Peldszus, and Stede, M. (2013). From Argument Diagrams to Argumentation Mining in Texts: A Survey. *Int. J of Cognitive Informatics and Natural Intelligence* 7(1), 1-31.
- Andrei Popescu-Belis (2005). *Dialogue Acts: One or More Dimensions?* Tech Report ISSCO Working paper n. 62.
- Vladimir Popescu, Jean Caelen, Corneliu Burileanu. *Logic-Based Rhetorical Structuring for Natural Language Generation in Human-Computer Dialogue. Lecture Notes in Computer Science Volume 4629*, pp 309-317, 2007.
- Dragomir R. Radev, Hongyan Jing, and Malgorzata Budzikowska. (2000). Centroid-based summarization of multiple documents: sentence extraction, utility-based evaluation, and user studies. In *Proceedings of the 2000 NAACL-ANLPWorkshop on Automatic summarization - Volume 4*.
- Marta Recasens, Marie-Catherine de Marneffe, and Christopher Potts (2013). The Life and Death of Discourse Entities: Identifying Singleton Mentions. In *Proceedings of NAACL*.
- Verena Rieser, Oliver Lemon (2011). *Reinforcement Learning for Adaptive Dialogue Systems: A Data-driven Methodology for Dialogue Management and Natural Language Generation*. Springer Science & Business Media, Nov 23, 2011 - Computers - 256 pages.
- Kate Rohit, Y. W. Wong, and R. Mooney (2005). Learning to transform natural to formal languages. In *AAAI*, 2005.

- Soumya Santhosh, Jahfar Ali (2012). Discourse Based Advancement On Question Answering System. Journal on Soft Computing.
- Merel Scholman, Jacqueline Evers-Vermeul, Ted Sanders (2016). Categories of coherence relations in discourse annotation. Dialogue & Discourse, Vol 7, No 2.
- Richard Socher, C. D. Manning, and A. Y. Ng (2010). Learning continuous phrase representations and syntactic parsing with recursive neural networks. In Proceedings of the NIPS-2010 Deep Learning and Unsupervised Feature Learning Workshop.
- Sparck Jones, K. (1995). Summarising: analytic framework, key component, experimental method', in Summarising Text for Intelligent Communication, (Ed. B. Endres-Niggemeyer, J. Hobbs and K. Sparck Jones), Dagstuhl Seminar Report 79, 13.12-17.12.93 (9350).
- Deirdre Wilson and Dan Sperber (2004). Relevance: Communication and Cognition. Blackwell, Oxford and Harvard University Press, Cambridge, MA,.
- Rajen Subba, and Barbara Di Eugenio (2009). An effective discourse parser that uses rich linguistic information Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics
- Mihai Surdeanu, Thomas Hicks, and Marco A. Valenzuela-Escarcega. (1986). 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), 2015.
- Zhao Tiancheng, Allen Lu, Kyusong Lee and Maxine Eskenazi (2017) Generative Encoder-Decoder Models for Task-Oriented Spoken Dialog Systems with Chatting Capability. SemDial 2017 Saarbrücken Germany.
- David R. Traum, and James F. Allen. (1994). Discourse obligations in dialogue processing. In Proceedings of the 32nd annual meeting on Association for Computational Linguistics (ACL '94). Association for Computational Linguistics, Stroudsburg, PA, USA, 1-8.
- David R. Traum, Hinkelman, Elizabeth A. (1992). Conversation Acts in Task-Oriented Spoken Dialogue. Computational Intelligence, 8(3), 575–599.
- Thomas Visser, David Traum, David DeVault, Rieks op den Akker (2014). A Model for Incremental Grounding in Spoken Dialogue Systems to appear in Journal of Multimodal User Interfaces,.
- David Vogel, Steffen Bickel, Peter Haider, Rolf Schimpfky, Peter Siemen, Steve Bridges, and Tobias Scheffer (2005). Classifying search engine queries using the web as background knowledge. SIGKDD Explor. Newsl. 7, 2 (December 2005), 117-122.
- W. Wang, Su, J., Tan, C.L. (2010). Kernel Based Discourse Relation Recognition with Temporal Ordering Information. ACL.
- Wikipedia (2016). Malaysia_Airlines_Flight_17.
https://en.wikipedia.org/wiki/Malaysia_Airlines_Flight_17.
- Y. A. Wilks (Ed.) (1999). Machine conversations. Kluwer.

8 Appendix

In Appendix we present a detailed chart for the Rhetorical Agreement algorithm with the references to the integrated components.

1. Define positive and negative classes of RR pairs:
 - a) Form the positive class from the rhetorically correct RR pairs
 - b) Form the negative class from the relevant but rhetorically foreign RR pairs
2. For each RR pair:
 - a) Parse each sentence

Stanford NLP Parser, NER, Sentiment module of (Manning et al., 2014, Recasens et al., 2013, Lee et al 2013)

b) Obtain VerbNet structure for verbs
VerbNet, JVerbNet (Kipper et al., 2008,
<http://projects.csail.mit.edu/jverbnet/>).

c) Obtain coreferences
Stanford NLP Parser – Coreference

d) Obtain entity - entity and entity – sub-entity links
OpenNLP.Similarity.parse_thicket

e) Build parse thicket pair for PT_{RR}
f) Apply discourse parsing to obtain discourse tree pair DT_{RR} for RR pair
g) Align EDUs of DT_{RR} with PT_{RR}
h) Merge aligned EDUs of DT_{RR} with PT_{RR}
OpenNLP.Similarity.parse_thicket

i) Obtain DT_{RR} with VerbNet signatures for CAs
j) Obtain parse thicket with enriched RST relations
OpenNLP.Similarity.parse_thicket.rhetoric_structure
k) Build representation for Thicket Kernel learning
l) Build representation for Nearest Neighbor learning
OpenNLP.Similarity.parse_thicket

m) Improve text similarity assessment by word2vec model
Mikolov et al., 2011, <https://deeplearning4j.org/>

3. Apply Thicket Kernel learning
OpenNLP.Similarity.parse_thicket.kernel_interface
Moschitti 2006, <http://disi.unitn.it/moschitti/Tree-Kernel.htm>

4. Apply Nearest Neighbor learning
OpenNLP.Similarity.jsmllearning
OpenNLP.Similarity.parse_thicket.matching

Fig. 10: Sources of Components for the Rhetorical Agreement classifier. References are shown in italics.