Computational Linguistics and Intellectual Technologies:
Proceedings of the International Conference “Dialogue 2017”

Moscow, May 31—June 3, 2017

LEARNING NOISY DISCOURSE TREES

Galitsky B. (boris.galitsky@oracle.com)

Oracle Corp Redwood Shores CA USA

It is well known that syntax-level analysis of user-generated text such as tweets and forum postings is unreliable due to its poor grammar and incompleteness. We attempt to apply a higher level linguistic analysis of rhetoric structure and investigate the potential application domains. We leverage an observation that discourse-level structure can be extracted from noisy text with higher reliability than syntactic links and named entities. As noisy text frequently includes informal interaction between agents, discussions, negotiations, arguments, complaints, we augment discourse trees with speech acts. Speech Act discourse tree (SADT) is defined as a discourse tree with verbs for speech acts as labels for its arcs. We identify text classification tasks which relies on tree kernel learning of SADTs: detection of negative mood (sentiment), text authenticity and answer appropriateness for question answering in social domains. The results are that the proposed technique outperforms on the discourse level traditional keyword-based algorithms in all of these three tasks.

Keywords: Speech Act Discourse Tree, sentiment analysis, answer appropriateness, text authenticity

1. Introduction

It is well known that text related to social network domains, what is called a user-generated data, is noisy. Therefore application of traditional natural language methods to texts written by non-professionals gives lower accuracy. One can expect that while processing noisy data (Jørgensen et al. 2015), certain level of generalization and abstraction would be beneficial. Similarly to other media such as images, an ascent to a higher-level of analysis would be fruitful.

In the last couple of years, availability of parsers which produce discourse structure significantly improved. Discourse parsers allows for an efficient automated analysis of rhetoric structures of text (Webber 2012, Joty et al 2013, Feng and Hirst 2014, Surdeanu et al 2015). Accuracy of discourse representations of rhetoric parsers has significantly improved, so that obtained discourse trees can be a subject of further automated analysis. However, a corpus of studies on applications of computed discourse trees is rather limited. In this study we explore how high-level discourse analysis of noisy text can be leveraged by a number of applications where traditional NLP techniques are fairly limited.

Marcu (1998) regarded a document as a Rhetorical Structure Theory (RST) (Mann & Thompson, 1988)-based discourse tree and selected textual units according to a preference ranking derived from the tree structure to make a summary. Representing an essence of a noisy text can be viewed from the text summarization perspective. Recent studies on text summarization formulate it as a combinatorial optimization problem, extracting the optimal subset from a set of the textual units that maximizes an objective function without violating the length constraint. Although these methods successfully improve automatic evaluation scores, they do not consider the discourse structure in the source document. To be logically coherent, (Hirao et al 2015) proposed a method that exploits a discourse tree structure to produce coherent summaries, transforming a traditional discourse tree, namely a rhetorical structure theory-based discourse tree, into a dependency-based discourse tree.

Chat bots frequently rely on ad-hoc solutions for the units making chat turn decisions (Popescu 2007). However, with the advent of novel dialogue planning techniques, integrating task-specific and general world knowledge in order to provide a more reliable and natural interaction with humans, more sophisticated chatbot response generation techniques are necessary. The authors present performance improvements employing a module that simplemented Segmented Discourse Representation Theory for response generation for chatbots, using the first-order logic (FOL) formalism, enforced by a task-independent discourse ontology. These improvements concern reductions in computational costs and enhancements in rhetorical coherence for the discourse structures obtained, and are obtained using speech-act related information for driving rhetorical relations computations.

Although discourse parsers rely on syntactic information, we expect them to perform reasonably well even when this information such as part-of-speech tags and syntactic trees are incomplete and noisy (van der Wees et al 2015). To further overcome this noisiness problem, we extend discourse trees with speech acts extracted from text to better represent the structure of what noisy text authors communicate and in which way.

Notice that slightly different texts might produce rather different DTs, and conversely, totally unrelated texts can produce the same DT even if the flow of rhetoric relations is not fully identical. DT parsers produce differences, which are not necessarily anchored in true discourse facts. Speech Act-based discourse trees (SADT) help to overcome this problem, since the traditional DTs are enriched with communicative discourse so that even if relations are misrepresented due to falsely fired syntactic rules, the structure of communication is still retained. We combine Speech Act Theory (Searle 1969) and Rhetoric Structure Theory. SADTs, in addition to relation between fragments of texts connected with rhetoric relations (Mann et al 1992), have special labels related to speech acts used by participants of a scenario to present a given rhetoric relation to the reader of the text.

Noisy discourse tree appear in the following tasks:

- Detecting a *logical argument* in text. A number of text genres of noisy text include argumentation, where an author attempts to back up her claim with certain statement. Argumentation is frequently associated with heated discussion. Conversely, multiple genres such as fact sharing, instructions and others do not include argumentation. In a content management system, it is important to automatically relate a noisy text to either the class of opinionated texts (with argumentation) or to the unbiased class (without argumentation).
- *Sentiment analysis*. Traditional, semantic compositionality methods of sentiment analysis are unreliable even when the text is not noisy. A user mood such as negative sentiment can be inferred from such paragraph-level features as intense argumentation, complex mental states such as deception, and others. Customer reviews and opinionated text is a good source of noisy text to explore how sentiment polarity can be inferred from the discourse-level features, since the lower-level linguistic features are rather noisy and unreliable.
- Text authenticity (validity, soundness, proper communication, confidence). It is rather harder to assess style-related features of a grammatically incorrect text based on its syntactic features. The degree of grammar deviation from normal is not a good indicator of content validity. It is hard to form explicit rules for how text style corresponds to its validity, therefore a supervised learning approach seems to be more plausible. An interesting and systematic example here are customer complaints, where the task of a customer support agent is to differentiate between
    1) valid, sound complaints requiring attention, from
    2) invalid, fake ones where a user is in a bad mood or just intends to receive a compensation.
- *Answer appropriateness commenting on a user post.* This is a special case of question answering, an automated support of user conversation, where the seed (the question or a request) is an incomplete or grammatically incorrect paragraph of text. To support a dialogue, a conversational agent needs to extract a topic from a seed and also maintain the coordination between the seed and response.

In all these domains, the problem is formulated as text classification into two classes:

- Positive (sentiment, authentic / valid text, correct answer or reply);
- Negative (sentiment, incorrect / invalid / incohesive text, incorrect answer or reply).

For a text to be classified into one of these classes, it has to be similar to its elements. We use statistical learning of structures with implicit feature engineering in the form of kernel learning of discourse trees as a reduction of a set of parse trees for a paragraph. If the solution to these problems for noisy text is satisfactory we can expect a broader range of application based on SADTs.

It turns out that using only rhetoric relations or only speech acts gives insufficient accuracy, but the combination of these sources produce acceptable results. More detailed syntactic and discourse information might help but can be redundant as well. In this study we will rely on information obtained from rhetoric relations and speech acts and compare the results with a classification system employing the syntactic data only.

If a text is shorter than a paragraph, such as Twitter, discourse-level analysis is believed to be inappropriate.

2. Representing a purpose of text in its DT

Conducting content exploration via chat bots or search engines (Galitsky 2013), discourse analysis is expected to help shortlisting answers are coordinated with a question in terms of style. The way an answer is communicated should be coordinated with the way a question is formulated. For example, if a user asking a question is a specialist in a certain adjucent area, an answer should contain a link between this specialty area and the focus of the question. The role of achieving agreement between user questions and user answers is especially high in noisy text domain.

Discourse-level agreement demands that A matches Q with respect to a domain knowledge and confidence, argumentation style, a level of politeness and other text features other than topics. On the other hand, discourse-level considerations are applied to Q/A topicality as well. If Q is represented as a sequence of keywords, and A is represented as a DT, then it is possible to formulate a simple rule-based system to filter out irrelevant answers based on how query keywords are distributed through the DT-A. These rules can be considered as constraints for the mapping between the nodes of trees

DT-Q → DT-A,

where PT(Q) is a trivial tree, a chain of words (we remove all edges and add unlabeled edges to link the nodes for words in a sequence); and DT-A is a tree with nodes for words and edges for rhetoric relations (all other edges are removed). Once an answer text is split into elementary discourse units (EDUs), and rhetoric relations are established between them, we establish rules for whether query keywords occurring in text are connected by rhetoric relations (and therefore this answer is likely relevant) or not connected (and this answer is most likely irrelevant). Hence we use a discourse tree (DT) as a base to identify certain sets of nodes in the DT to corresponding to Qs so that this text A is a valid answer, and certain sets of nodes correspond to invalid answers.

Usually, the main clause of a multi-sentence question includes the main entity of Q and some of its attributes, and supplementary clauses include other attributes and possibly constraints on them. In the most straight-forward way, the main clause of a question is mapped into a nucleus, and the supplementary clause is mapped into a satellite of the RST relation, such as elaboration. Linkage by other RST relations, where a satellite introduces additional constraints for a nucleus, has the same meaning for answer validity. This validity still holds where two EDUs are connected with a symmetric relation, such as joint. However, when the images of the main and supplementary clause of Q are satellites of *different* nuclei in A, it most likely means that they express constraints for different entities and therefore this A is irrelevant for this Q.

We start with an example of answer text split into EDUs:

[furthermore,] e1 [they think] e2 [stock volatility maximum is not occurring at the same time in the past,] e3 [because of production and pricing differences] e4 [that are limiting the accuracy of seasonal adjustments] e5 [built into the financial data.] e6

A DT including 6 nodes {e1...e6} is shown in Fig. 0. Horizontal lines indicate text segments; satellites are connected to their nuclei by curved arrows.

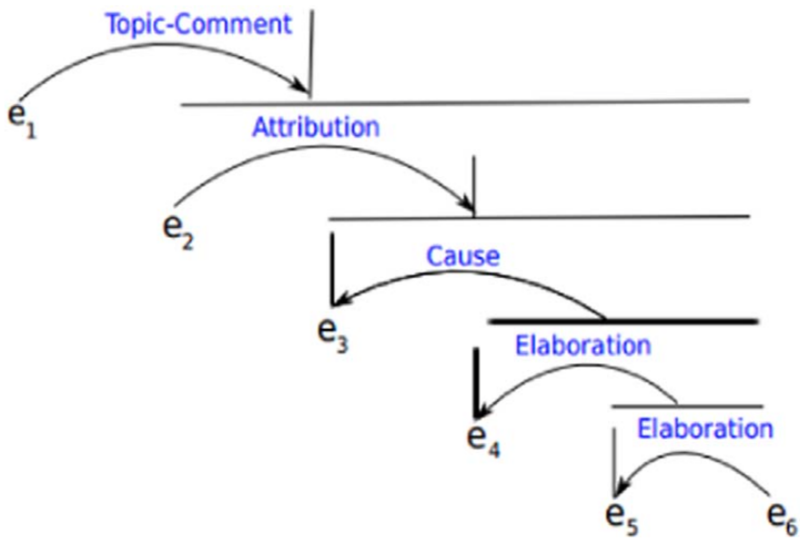


Fig. 0. Initial example of a DT

One can see that this text is a relevant answer for the question

Are seasonal swings in stock price volatility due to pricing differences?

because the respective areas e3 and e4 in the DT-A are ({*stock, volatility, maximum, …, due, pricing, differences*}) → DT-Q({… e3, e4, …}), {*seasonal, swings*} → e3, {*pricing, differences*} → e4. However, this answer is irrelevant for the question

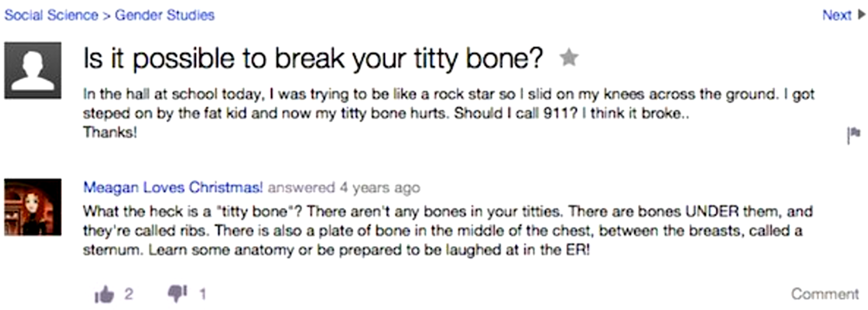
Are pricing differences built into employment data?

because the areas e4 and e6 in the rhetoric map of the answer are not connected. EDU e6 is an elaboration of e5, and e5 is, in turn, an elaboration of e4; however, e4 and e6 are not logically connected and cannot be mapped into by a set of question keywords.

3. Mapping DT-Q into DT-A

We introduce an example of a question and its answer (CollegeHumor 2017) and show that their DTs have to agree. We will demonstrate that SDT is an andequate means to express this form of agreement. If a question has a certain logic expressed by a discourde structure, the answer has to match it in some way. Q/A pair and the respective pair of discourse trees is shown in Fig. 1 (Q is on the left and A is on the right).

The main contradiction in this Q/A pair is that the Q demonstrates a lack of knowledge on a subject and A includes an argument that this knowledge needs to be acquired. Relation *contrast* in Q has to be addressed in A. Since Q is asking whether an accident is serious (and a trip to emergency room is necessary) or not, A has to include this relation, considering cases when the Q author is knowledgable in anatomy or not and how it affects the emergency room visit. Hence we map *Q-contrast* into *A-contrast*. Also, the *elaboration* relation associated with *Q-contrast* is mapped into *elaboration* relation associated with *A-contrast*.



As to a formal definition of a SADT, it is as follows. SADT is a DT with labels for arcs that are the VerbNet expressions for verbs which are related to speech acts. The arguments of these verbs are substituted from text according to VerbNet frames. The first argument is instantiated by an agent and the second by a noun phrase that is a subject of a speech act. Further details on DTs are available in (Joty et al 2016), and on VerbNet Frames—in (Kipper et al 2008).

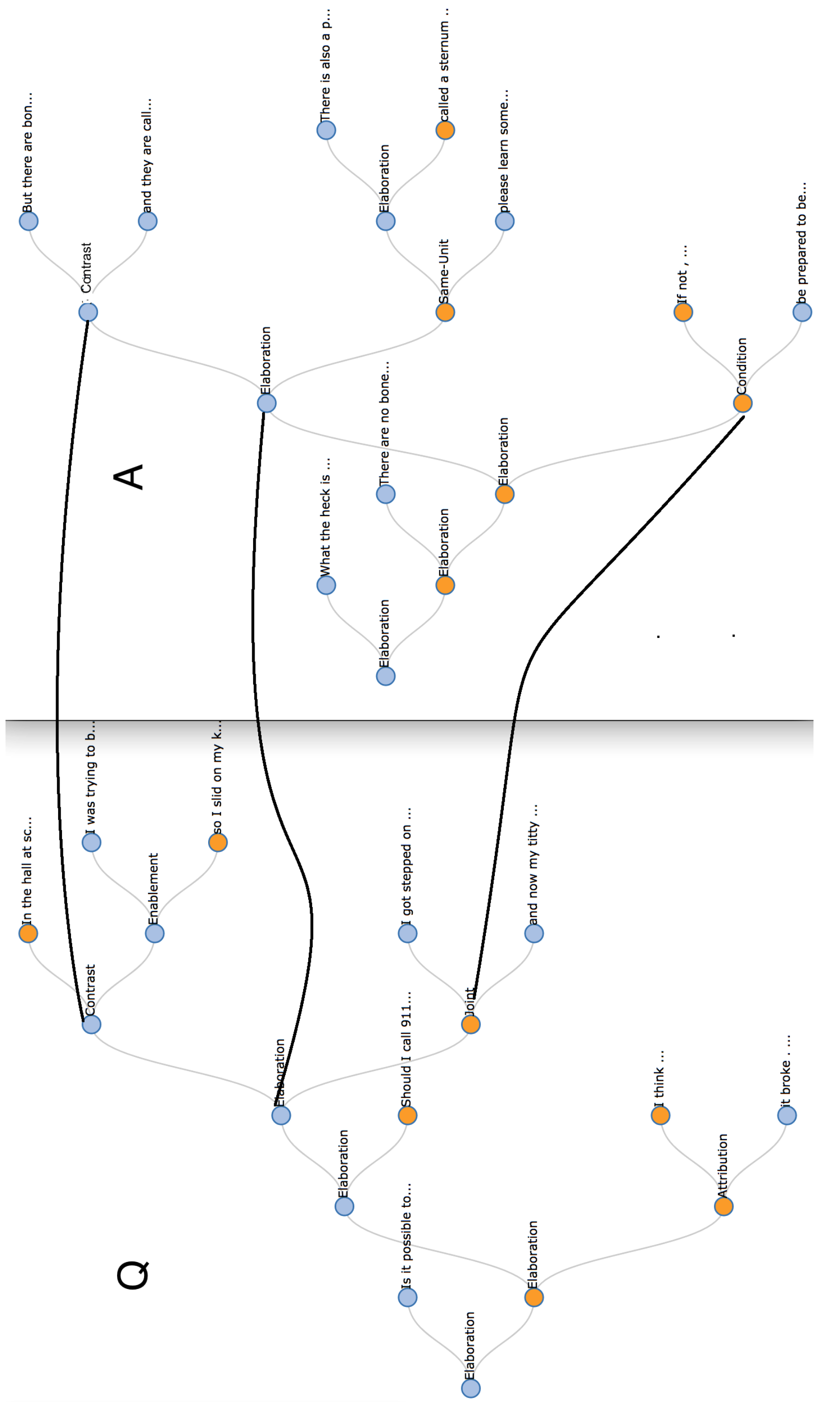


Fig. 1. Discourse Trees for a question and an answer have to be coordinated

4. Similarity function for learning SADT

Deep learning approach is not well suited to be applied to structured data since feature engineering and explainability are difficult. Deep learning can potentially apply a more complex feature space and assure a higher classification accuracy, but does not help in understanding or exploring the phenomena. We therefore use inductive and statistical approaches:

1) Represent SADTs in a numerical space, and express similarity as a number. This is a tree-kernel approach that belongs to statistical learning family. The feature space includes all SADTs sub-trees.
2) Use a structural representation, without numerical space, such as trees and graphs, and express similarity as a maximal common sub-structure (Galitsky 2012, 2016). We refer to such operation as *similarity operation* (*generalization*, ‘^’). This is an inductive learning approach.

We use the former approach to assess how text classification tasks can be consistently handled when the data becomes noisier and syntactic analysis produces more errors. The latter one is superior in terms of feature engineering but is also less universal and would need special representations of DTs depending on an application area. Therefore a hybrid approach combing best of both worlds would be beneficial (not evaluated in this study).

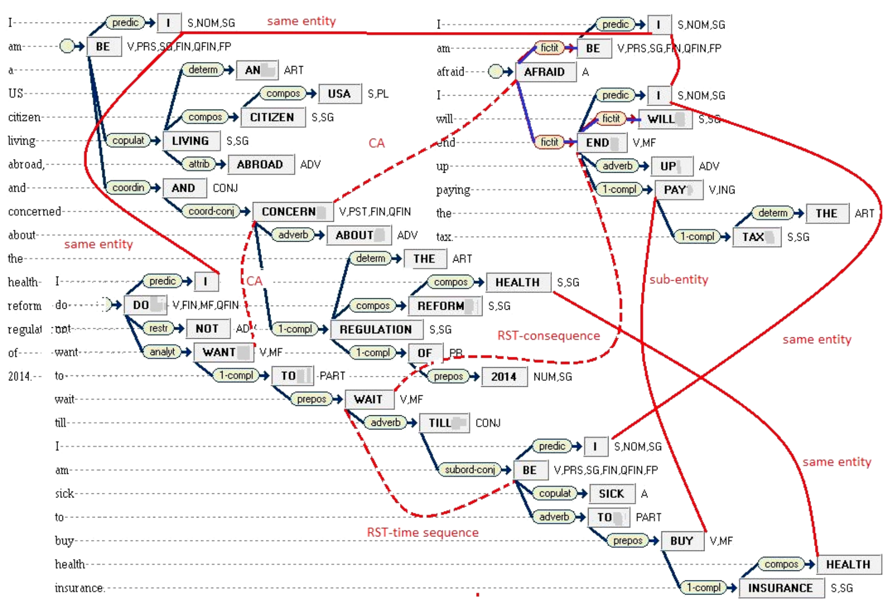


Fig. 2: A Parse thicket for a question

(Galitsky et al 2015) combined parse trees for sentences with discourse-level relationships between words and parts of the sentence in one graph, called *parse thicket*. The straight edges of this graph are syntactic relations, and curvy arcs—discourse relations, such as anaphora, same entity, sub-entity, rhetoric relation and communicative actions. Fig. 2 shows the parse thicket for the question *I am a US citizen living abroad, and concerned about the health reform regulation of 2014. I do not want to wait till I am sick to buy health insurance. I am afraid I will end up paying the tax.*

Parse thicket includes much more complete information than just a combination of parse trees for individual sentences would, especially when these trees are noisy. Navigation through the parse thicket along the edges for syntactic relations as well as the arcs for discourse relations allows one to transform a given parse thicket into semantically equivalent forms for matching with other parse thickets, performing a text similarity assessment task at the level of paragraph, irrespectively how it is split into sentences. Parse thickets also help to do relevance assessment with noisy text where syntactic analysis is subject to numerous errors and omissions. SADT is a subtree of parse thicket as a graph with the focus on rhetoric-level information only.

4.1. Tree Kernel learning for SADT

Tree Kernel learning for strings, parse trees and parse thickets is a well-established research area nowadays. The parse tree kernel counts the number of common sub-trees as the discourse similarity measure between two SADTs. Tree kernel has been defined for DT by (Joty and Moschitti 2014). (Wang et al 2013) used the special form of tree kernels for discourse relation recognition. In this study we extend the tree kernel definition for the SADT, augmenting DT kernel by the information on communicative actions. A SADT can be represented by a vector of integer counts of each sub-tree type (without taking into account its ancestors).

We combined Stanford NLP parsing, coreferences, entity extraction, DT construction (discourse parser, Surdeanu et al 2013 and Joty et al 2016), VerbNet and Tree Kernel builder into one system available at https://github.com/bgalitsky/relevance-based-on-parse-trees.

5. Evaluation of SVM TK learning of SADT in four domains

To detect sentiments, we first need to learn to detect a mixture of opinions, a conflict, a presence of logical argumentation in text. Then we build a hybrid sentiment classification system relying upon the detected cases of opposing argumentation.

5.1. Detecting noisy argumentation

We formed the *positive* dataset from the noisy text data where argumentation is frequent, e.g. opinionated letters to the editors of major US newspapers. We also used textual customer complaints dataset from our previous evaluations. Besides, we use the text style & genre recognition dataset (Lee, 2001) which has a specific dimension associated with argumentation. For the *negative* dataset, we used a non-noisy text sources such as Wikipedia and factual news sources. Both datasets include 3600 texts.

Table 1: Evaluation results for detecting logical argument

Method / sources	Newspaper opinions			Customer complaints		
	Precision	Recall	F-measure	Precision	Recall	F-measure
Naïve Bag-of-words	63.4	56.7	59.86	52.3	54.2	53.23
WEKA-Naïve Bayes	64.7	57	60.61	56.7	52.6	54.57
SVM TK for RST and SA (full parse trees combined in parse thicket)	78.8	72.9	75.74	74.6	70.2	72.33
SVM TK for DT (w/o SA)	62.4	61.7	62.05	59.3	63.2	61.19
SVM TK for SADT	81.9	76.3	79.00	75.2	74.6	74.90

SVM TK baseline is shown as light-greyed area in the middle row of Table 1. Representation includes exhaustive syntactic information in the form of parse thickets. The best algorithm of the current study, SVM TK for SADT (bottom greyed row) outperforms SVM TK for traditional DTs (without speech acts) by as much as 25% and full-set syntactic features (the SVM TK baseline) by only 3%. We conclude that contribution of speech act—related information for noisy text is substantial. A small gain in accuracy is due to the fact that noisy text syntactic data is noisy, and its addition decreases the recognition accuracy instead of increasing it.

5.2. Improvement of sentiment detection

Since reliable sentiment detection in an arbitrary domain is extremely hard, we focus on a particular sentiment—related feature such as logical argumentation and observe how its detection (Section 4.1) can help overall sentiment assessment. We formulate sentiment detection problem for noisy text at the level of paragraphs, only detecting sentiment polarity. For evaluation, we use a dataset of positive and negative, genuine and fake travelers review of Chicago area hotels (Ott et al 2013).

The results of sentiment analysis achieved by the hybrid compositional semantics and discourse analysis are shown in Table 2. In the first row we show the accuracy of the baseline system on our data. In the second grayed row we show the improvement by means of the hybrid system. This improvement of almost 15% is achieved by discovering overall negative sentiment at the paragraph level in case of recognized presence of argumentation. In some of these cases the negative sentiment is implicit and can only be detected indirectly from the discourse structure, where individual words do not indicate negative sentiments.

Table 2. Evaluation of sentiment analysis task

Data source and method	Precision	Recall	F-measure
Baseline sentiment detector (Standord NLP Sentiment)	62.7	68.3	65.38
Hybrid sentiment detector (Stanford NLP + SVM TK for SADT)	79.3	81.0	80.14
Sentiment detector via SVM TK for SADT	69.8	68.3	69.04

5.3. Accessing authenticity of customer complaints and reviews

Table 3. Evaluation of complaint/review validity task

Data source and method	F
Untruthful opinion data detector, *positive* reviews (SVM TK SADT)	77.26
Untruthful opinion data detector, *negative* reviews (SVM TK for SADT)	76.23
SVM TK of unconnected parse trees	62.84
SVM TK of parse thicket with anaphora only	65.10
SVM TK of with anaphora and Stanford sentiment profiles	74.46
SVM TK of parse thickets with anaphora and RST	78.98
SVM TK of SADT	80.03

We explored whether fake opinionated text have different rhetoric structure to genuine one (Table 3).

Although our SVM TK system did not achieve (Ott et al 2011, 2013) performance of 90%, the task of detection of fake review texts was performed (at 76–77% accuracy, two bottom greyed rows) by the universal text classification system, the same that extracts arguments, finds rhetorically suitable answers and assesses sentiments polarity. We also accessed the validity of customer complaints, based on the manually tagged set of a limited size. We observed how adding discourse information improves recognitions accuracy: we start with unconnected parse trees, then add anaphora and RST, and finally proceed to SADT (bottom of Table 3).

5.4. Assessing coordination a question and an answer

Our evaluation dataset included 560 Answers and Questions scraped from public sources. We consider the pair *Question-Best Answer* as an element of the positive training set and *Question-OtherAnswer* as the one of the negative training set.

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 is adhered to a question-or-request vs response format. Then the respective Q/A pair of texts is formed. The search is implemented via Bing Azure Search Engine API in the Web and News domains.

Answer classification accuracies are shown in Table 4. Each row represents a particular method; each class of methods in shown in grayed areas.

Table 4. Evaluation of the coordination task

Source / Evaluation setting	Community Answers		
	P	**R**	**F1**
Types and Counts for rhetoric relations of Q and A	55.2	52.9	54.03
Entity-based alignment of DT of Q and A	63.1	57.8	60.33
SVM TK for Parse Trees of individual sentences	66.1	63.8	64.93
SVM TK for RST and SA (parse thickets	75.8	74.2	74.99

Source / Evaluation setting	Community Answers		
	P	R	F1
SVM TK for RR-DT	76.5	77.0	76.75
SVM TK for RR-SADT	80.3	78.3	79.29
SVM TK for RR-SADT + sentiment + argumentation features	78.3	76.9	77.59

Our evaluation settings are close to SVM-based ranking of RST parses. The rhetoric relevance recognition accuracy is also comparable with the state of art in question answering systems relying on rhetoric features such as (Jansen et al. 2013).

Conclusion

In this study we defined SADT and proposed a statistical SVM TK based learning framework that can be applied to a manifold of NLP tasks. SADT allows combining the structure of rhetoric relation with the structure of communication, which complements each other being applied to noisy noisy text.

Using SVM TK one can differentiate between a broad range of styles of noisy text (Galitsky et al 2015). Each text style and genre has its inherent rhetoric structure that is leveraged and automatically learned. When syntactic structure is noisy and some features can be missing, the rhetoric structure with unreliably detected EDUs can still be a reliable indicator of text style. 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 the accuracy in the complex cases.

An extensive corpus of literature on RST parsers does not address the issue of how the resultant DT will be employed in practical NLP systems. RST parsers are mostly evaluated with respect to agreement with the test set annotated by humans rather than its expressiveness of the features of interest. In this work we focused on interpretation of DT for noisy text and explored ways to represent them in a form indicative of a conflict, negative sentiment, sharing of authentic information rather than neutral enumeration of facts.

We demonstrated that this discourse-level technique performs better than traditional keyword-based statistical and/or compositional semantics approaches in all of these four tasks. We also showed that this improvement is larger for user-generated content in comparison with the professionally written text with proper style and grammar. Classification of SADTs gives a higher accuracy than a conventional sentiment analysis. Text validity assessment for a gives satisfactory results, comparable to general style classification accuracies obtained elsewhere. Also, rhetoric support for answer relevance demonstrated the accuracies comparable with the state-of-the-art for community question answering (Jansen et al 2013).

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

References

1. *Lee, D.* (2001) Genres, registers, text types, domains and styles: Clarifying the concepts and navigating a path through the BNC jungle. Language Learning & Technology, Vol. 5(3): 37–72.
2. *Collegehumor.com* (2017). http://www.collegehumor.com/post/7013323/22-yahoo-answers-questions-that-just-might-rot-your-brain/page:3. Last downloaded Feb 10, 2017.
3. *Mann William and Sandra Thompson.* (1988) Rhetorical structure theory: Towards a functional theory of text organization. Text-Interdisciplinary Journal for the Study of Discourse, 8(3):243–281.
4. *Webber, B., Mark Egg, and Valia Kordoni.* (2012) Discourse structure and language technology. Natural Language Engineering, 18:437–490.
5. *Marcu, D.* (1998) Improving summarization through rhetorical parsing tuning. In Proc. of the 6th Workshop on Very Large Corpora, pages 206–215.
6. *WenTing Wang, Su Jian, Chew Lim Tan.* (2010). Kernel Based Discourse Relation Recognition with Temporal Ordering Information. ACL.
7. *Feng, Vanessa Wei and Graeme Hirst.* (2014) A linear- time bottom-up discourse parser with constraints and post-editing. In Proceedings of The 52nd An- nual Meeting of the Association for Computational Linguistics (ACL 2014), Baltimore, USA, June.
8. *Joty, S., G. Carenini, R. T. Ng, and Y. Mehdad.* (2013) Combining intra-and multi-sentential rhetorical parsing for document-level discourse analysis. In ACL (1), pages 486–496.
9. *Joty, S. and A. Moschitti.* (2014) Discriminative Reranking of Discourse Parses Using Tree Kernels. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP).
10. *Joty, Shafiq R., Giuseppe Carenini, Raymond T Ng.* (2016) CODRA: A Novel Discriminative Framework for Rhetorical Analysis. Computational Linguistics Volume 41, Number 3, 2016.
11. *Jørgensen, Anna, Dirk Hovy and Anders Søgaard.* (2015) Challenges of studying and processing dialects in social media. Proceedings of the ACL 2015 Workshop on Noisy User-generated Text.
12. *P. Jansen, M. Surdeanu, and P. Clark.* (2014) Discourse Complements Lexical Semantics for Nonfactoid Answer Reranking. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL).
13. *Galitsky B., Ilvovsky D., Kuznetsov S. O.* (2015) Text Classification into Abstract Classes Based on Discourse Structure, in: Proceedings of the Recent Advances in Natural Language Processing, RANLP 2015. Hissar. P. 201–207.
14. *Galitsky, B.* (2012) Machine Learning of Syntactic Parse Trees for Search and Classification of Text. Engineering Application of AI.
15. *Galitsky, B.* (2016) Generalization of parse trees for iterative taxonomy learning. Information Sciences. Volume 329, n1 February 2016, Pages 125–143.
16. *Galitsky B.* (2013) Content inversion for user searches and product recommendations systems and methods. US Patent 9336297. https://www.google.com/patents/US9336297.

17. *Kipper K., Korhonen A., Ryant N. and Palmer M.* (2008) A large-scale classification of English verbs. Language Resources and Evaluation Journal, 42, pp.21–40.
18. *Popescu, Vladimir, Jean Caelen, Corneliu Burileanu* (2007) Using Speech Acts in Logic-Based Rhetorical Structuring for Natural Language Generation in Human-Computer Dialogue. Text, Speech and Dialogue, 10th International Conference, TSD 2007, Pilsen, Czech Republic, September 3–7, 2007, Proceedings
19. *Searle, J.* (1969). Speech acts: An essay in the philosophy of language. Cambridge: Cambridge University Press.
20. *Surdeanu, Mihai, Thomas Hicks, and Marco A. Valenzuela-Escarcega.* (2015) Two Practical Rhetorical Structure Theory Parsers. Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics—Human Language Technologies: Software Demonstrations (NAACL HLT).
21. *Ott, M., 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.
22. *Ott, M., 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.
23. *Jindal, N., and Liu, B.* (2008) Opinion spam and analysis. In WSDM, 219–230.
24. *Marlies van der Wees, Arianna Bisazza and Christof Monz.* (2015) Five Shades of Noise: Analyzing Machine Translation Errors in User-Generated Text. Proceedings of the ACL 2015 Workshop on Noisy User-generated Text.
25. *Hirao, T., Masaaki Nishino, Yasuhisa Yoshida, Jun Suzuki, Norihito Yasuda, and Masaaki Nagata.* (2015) Summarizing a document by trimming the discourse tree. IEEE/ACM Trans. Audio, Speech and Lang. Proc. 23, 11 (November 2015), 2081–2092. DOI=http://dx.doi.org/10.1109/TASLP.2015.2465150