Text Classification into Abstract Classes Based on Discourse Structure

Boris Galitsky
Knowledge Trail Inc.
San-Francisco, USA
bgalitsky@hotmail.com

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

Sergey O. Kuznetsov
National Research University Higher School of Economics, Moscow, Russia
skuznetsov@hse.ru

Abstract

The problem of classifying text with respect to belonging to a document or a meta-document is formulated and its application areas are proposed. An algorithm is proposed for document classification tasks where counts of words is insufficient do differentiate between such abstract classes of text as metalanguage and object-level. We extend the parse tree kernel method from the level of individual sentences towards the level of paragraphs, based on anaphora, rhetoric structure relations and communicative actions linking phrases in different sentences. Tree kernel learning technique is applied to these extended trees to leverage of additional discourse-related information. We evaluate our approach in the domain of action-plan documents.

1 Introduction

Solving text classification problems, keywords and their topicality usually suffice. These features provide abundant information to determine a topic of a text or document, such as apple vs banana, or adventures vs relaxing travel. At the same time, there is a number of document classification domains where distinct classes have similar words. In this case, style, phrasings and other kinds of text structure information need to be leveraged. To perform text classification in such domains, one needs to employ discourse information such as anaphora, rhetoric structure, entity synonymy and ontology, if available (Wu et al., 2011).

In this study, an issue of classifying a text with respect to being metalanguage or language object is addressed. We are concerned with differentiating between object-level documents, which inform us on how to do things, or how something has been done, and meta-documents, specifying how to write a document which explains how to do things, or how things have been done. Metalanguage is a symbolic system intended to express information, or analyze another language or symbolic system. In a natural language document, metalanguage is used as a special expressive means to ascend to the desired level of abstraction. To automatically recognize metalanguage patterns in text, one needs some implicit signals at the syntactic level. Naturally, just using keyword statistics is insufficient to differentiate between texts in metalanguage and language-object.

A presence of verbs for speech acts and mental states (such as knowing) may help to identify metalanguage patterns, but is an unreliable criterion: *I know the location of the highest mountain* vs *I know what he thinks about the highest mountain in the world*. The latter sentence contains a meta-predicate *think (who, about-what)* with the second variable ranging over a set of (object-level) expressions for thoughts about the *highest mountain*. Relying on syntactic parse trees would provide us with specific expressions and phrasings connected with a metalanguage. However, it will still be insufficient for a thorough description of linguistic features inherent to a metalanguage. It is hard to identify such features without employing a discourse structure of a document. This discourse structure needs to include anaphora, rhetoric relations, and interaction scenarios by means of communicative language (Galitsky and Kuznetsov, 2008). Furthermore, to systematically learn these discourse features associated with metalanguage, and differentiate them from the ones for language-object, one needs a unified approach to classify graph structures at the level of paragraphs (Galitsky et al., 2013).

The design of such features for automated learning of syntactic and discourse structures for classification is still done manually today. To overcome this problem, tree kernel approach has been proposed (Cumby and Roth, 2003). Tree kernels constructed over syntactic parse trees, as well as discourse trees (Galitsky et al., 2015) is one of the solutions to conduct feature engineer-

Proceedings of Recent Advances in Natural Language Processing, pages 200–207,
Hissar, Bulgaria, Sep 7–9 2015.

ing. Convolution tree kernel (Collins and Duffy, 2002; Haussler, 1999) defines a feature space consisting of all subtree types of parse trees and counts the number of common subtrees to express the respective distance in the feature space. They have found a broad range of applications in NLP tasks such as syntactic parsing re-ranking, relation extraction (Zhang et al., 2008), named entity recognition (Cumby and Roth, 2003), pronoun resolution (Kong and Zhou, 2011), question classification, and machine translation.

The kernel ability to generate large feature sets is useful to assure we have enough linguistic features to differentiate between the classes, to quickly model new and not well understood linguistic phenomena in learning machines. However, it is often possible to manually design features for linear kernels that produce high accuracy and fast computation time whereas the complexity of tree kernels may prevent their application in real scenarios. SVM (Vapnik, 1995) can work directly with kernels by replacing the dot product with a particular kernel function. This useful property of kernel methods, that implicitly calculates the dot product in a high-dimensional space over the original representations of objects such as sentences, has made kernel methods an effective solution to modeling structured linguistic objects (Moschitti, 2006).

An approach to build a kernel based on more than a single parse tree for search has been proposed (Galitsky et al., 2015). To perform classification based on additional discourse features, we form a single tree from a tree forest for a sequence of sentences in a paragraph of text.

A number of NLP tasks such as classification require computing of semantic features over paragraphs of text containing multiple sentences. Doing it at the level of individual sentences and then summing up the score for sentences will not always work. In the complex classification tasks where classes are defined in an abstract way, the difference between them may lay at the paragraph level and not at the level of individual sentences. In the case where classes are defined not via topics but instead via writing style, discourse structure signals become essential. Moreover, some information about entities can be distributed across sentences, and classification approach needs to be independent of this distribution. We will demonstrate the contribution of paragraph-level approach vs the sentence level in our evaluation.

2 Text Classification Based on Discourse Text Structure

2.1 The domain of documents and meta-documents

Our first example of the use of meta-language is the following text shared by an upset customer, doing his best to have a bank to correct an error: *The customer representative acknowledged that the only thing he is authorized to do is to inform me that he is not authorized to do anything.*

This is a good example for how people describe *thinking about thinking*. In this example, bank operations can be described in language-object, and bank employee's authorizations to perform these operations are actually described in meta-language. Here a document on banking operations is an object-level document, and authorization rules document is a meta-document relative to the operations document. The claim of this work is that this classification can be performed based on text analysis only without any knowledge of banking industry.

We define an ***action-plan*** (object-level) document as a document which contains a thorough and well-structured description of how to build a particular system or work of art, from engineering to natural sciences to creative art. According to our definition, action-plan document follows the reproducibility criteria of a patent or research publication; however format might deviate significantly. One can read such document and being proficient in the knowledge domain, can build such a system or work of art.

Conversely, a meta-document is a document explaining how to write object-level, action-plan documents. They include manuals, standard action-plan documents should adhere to, tutorials on how to improve them, and others.

We need to differentiate action-plan documents from the classes of documents which can be viewed as ones containing meta-language, whereas the genuine action-plan documents consists of the language-object patterns and should not include metalanguage ones. As to the examples of meta-documents, they include design requirements, project requirement document, operational requirements, design guidelines, design guides, tutorials, design templates (template for technical design document, research papers on system design, educational materials on system design, resume of a design professional, and others.

Naturally, action-plan documents are different from similar kinds of documents on the same topic in terms of style and phrasing. To extract these features, rhetoric relations are essential. Notice that meta-documents can contain object-level text, such as design examples. Object level documents (genuine action-plan docs) can contain some author reflections on the system design process (which are written in metalanguage). Hence the boundary between classes does not strictly separates metalanguage and language object. We use statistical language learning to optimize such boundary, having supplied it with a rich set of linguistic features up to the discourse structures. In the design document domain, we will differentiate between texts expressed mostly via meta-language and the ones mostly in language-object.

2.2 Discourse Structure of a Document

It turns out that sentence-level tree kernels are insufficient for classification in our domains. Since important phrases can be distributed through different sentences, one needs a sentence boundary – independent way of extracting both syntactic and discourse features. Therefore we intend to combine/merge parse trees to make sure we cover all the phrase of interest. Let us analyze the following text with respect of belonging to a document or meta-document.

This document describes the design of back end processor. Its requirements are enumerated below.

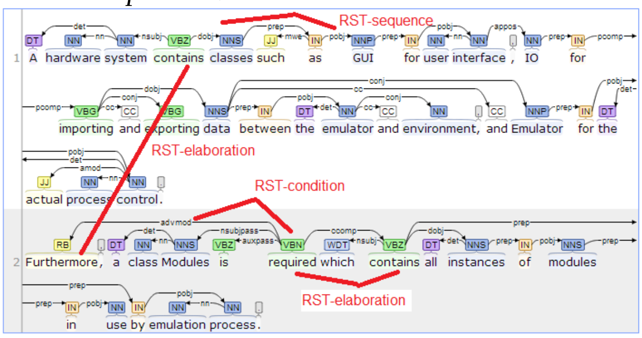
From the first sentence, it looks like an action-plan document. To process the second sentence, we need to disambiguate the preposition 'its'. As a result, we conclude from the second sentence that it is a requirements document, not an object-level action-plan one.

The structure of a document which can be potentially valuable for classification can be characterized by rhetoric relations that hold between the parts of a text. These relations, such as explanations or contrast, are important for text understanding in general since they contain information on how these parts of text are related to each other to form a coherent discourse. Naturally, we expect the structure of discourse for meta-language text patterns to be different to that of language-object text patterns.

Rhetorical Structure Theory, or RST (Mann, and Thompson, 1988; Mann et al., 1992; Marcu, 1997) is one of the most popular approaches to model extra-sentence as well as intra-sentence discourse. RST represents texts by labeled hierarchical structures, called Discourse Trees (DTs). The leaves of a DT correspond to contiguous Elementary Discourse Units (EDUs). Adjacent EDUs are connected by rhetorical relations (e.g., Elaboration, Contrast), forming larger discourse units (represented by internal nodes), which in turn are also subject to this relation linking. Discourse units linked by a rhetorical relation are further distinguished based on their relative importance in the text: nucleus being the central part, whereas satellite being the peripheral one. Discourse analysis in RST involves two subtasks: discourse segmentation is the task of identifying the EDUs, and discourse parsing is the task of linking the discourse units into a labeled tree. Discourse analysis explores how meanings can be built up in a communicative process, which varies between a text metalanguage and a text language-object. Each part of a text has a specific role in conveying the overall message of a given text.

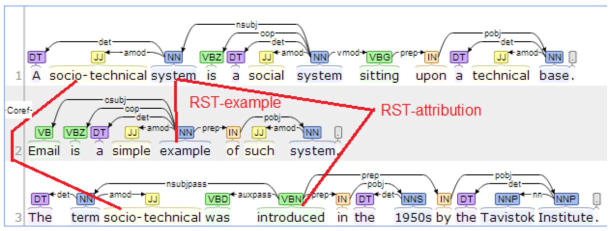
For our classification tasks, just an analysis of a text structure can suffice for proper classification. Given a positive sequence

A hardware system contains classes such as GUI for user interface, IO for importing and exporting data between the emulator and environment, and Emulator for the actual process control. Furthermore, a class Modules is required which contains all instances of modules in use by emulation process.



and a negative sequence

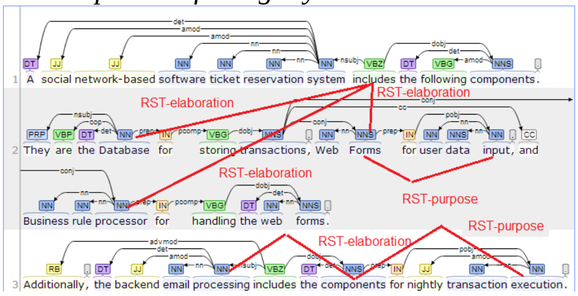
A socio-technical system is a social system sitting upon a technical base. Email is a simple example of such system. The term socio-technical was introduced in the 1950s by the Tavistok Institute.



We want to classify the paragraph

A social network-based software ticket reservation system includes the following components.

They are the Database for storing transactions, Web Forms for user data input, and Business rule processor for handling the web forms. Additionally, the backend email processing includes the components for nightly transaction execution.



One can see that it follows the rhetoric structure of the top (positive) training set element, although it shares more common keywords with the bottom (negative) element. Hence we classify it as an action-plan document, being an object-level text, since it describes the system rather than introduces a terms (as the negative element does).

2.3 Anaphora and Rhetoric Relations for Classification Task

We introduce a classification problem where keyword and even phrase-based features are insufficient. This is due to the variability of ways information can be communicated in multiple sentences, and variations in possible discourse structures of text which needs to be taken into account.

We consider an example of text classification problem, where short portions of text belong to two classes:

- Tax liability of a landlord renting office to a business.
- Tax liability of a business owner renting an office from landlord.

I rent an office space. This office is for my business. I can deduct office rental expense from my business profit to calculate net income.

To run my business, I have to rent an office. The net business profit is calculated as follows. Rental expense needs to be subtracted from revenue.

To store goods for my retail business I rent some space. When I calculate the net income, I take revenue and subtract business expenses such as office rent.

I rent out a first floor unit of my house to a travel business. I need to add the rental income to my profit. However, when I repair my house, I can deduct the repair expense from my rental income.

I receive rental income from my office. I have to claim it as a profit in my tax forms. I need to add my rental income to my profits, but subtract rental expenses such as repair from it.

I advertised my property as a business rental. Advertisement and repair expenses can be subtracted from the rental income. Remaining rental income needs to be added to my profit and be reported as taxable profit.

Note that keyword-based analysis does not help to separate the first three paragraph and the second three paragraphs. They all share the same keywords *rental/office/income/profit/add/subtract.* Phrase-based analysis does not help, since both sets of paragraphs share similar phrases.

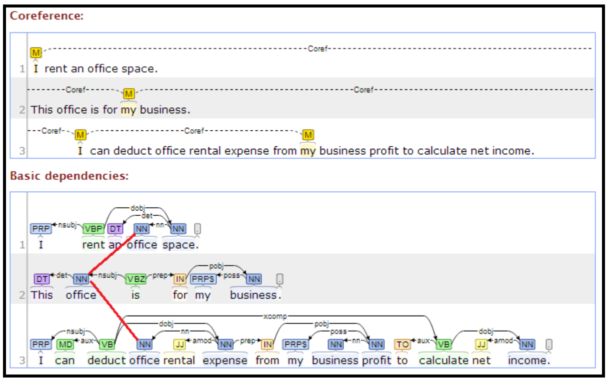
Secondly, pair-wise sentence comparison does not solve the problem either. Anaphora resolution is helpful but insufficient. All these sentences include '*I*' and its mention, but other links between words or phrases in different sentences need to be used.

Rhetoric structures need to come into play to provide additional links between sentences. The structure to distinguish between

renting for yourself and deducting from total income and

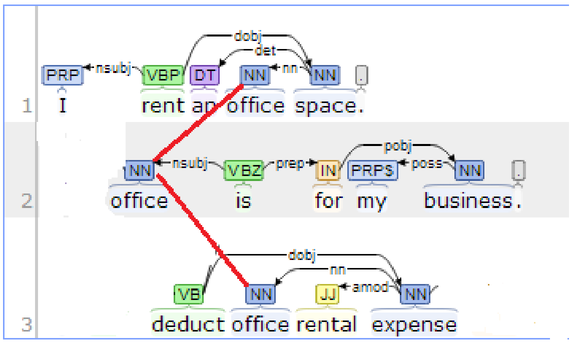
renting to someone and adding to income embraces multiple sentences. The second clause about *adding/subtracting incomes* is linked by means of the rhetoric relation of *elaboration* with the first clause for *landlord/tenant*. This rhetoric relation may link discourse units within a sentence, between consecutive sentences and even between first and third sentence in a paragraph. Other rhetoric relations can play similar role for forming essential links for text classification.

Which representations for these paragraphs of text would produce such common sub-structure between the structures of these paragraphs? We believe that extended trees, which include the first, second, and third sentence for each paragraph together can serve as a structure to differentiate the two above classes. The dependency parse trees for the first text in our set and its coreferences are shown below.



There are multiple ways the nodes from parse trees of different sentences can be connected: we choose the rhetoric relation of elaboration which links the same entity office and helps us to form the structure *rent-office-space – for-my-business – deduct-rental-expense* which is the base for our classification.

We show the resultant extended tree with the root *'I'* from the first sentence.



It includes the whole first sentence, a verb phrase from the second sentence and a verb phrase from the third sentence according to rhetoric relation of elaboration. Notice that this extended tree can be intuitively viewed as representing the 'main idea' of this text compared to other texts in our set. All extended trees need to be formed for a text and then compared with that of the other texts, since we don't know in advance which extended tree is essential. From the standpoint of tree kernel learning, extended trees are learned the same way as regular parse trees.

2.4 Learning on Extended Trees

For every inter-sentence arc which connects two parse trees, we derive the extension of these trees, extending branches according to the arc (Fig. 1).

In this approach, for a given parse tree, we will obtain a set of its extension, so the elements of kernel will be computed for many extensions, instead of just a single tree. The problem here is that we need to find common sub-trees for a much higher number of trees than the number of sentences in text, however by subsumption (sub-tree relation) the number of common sub-trees will be substantially reduced.

If we have two parse trees P_1 and P_2 for two sentences in a paragraph, and a relation R_{12}: $P_{1i} \rightarrow P_{2j}$ between the nodes P_{1i} and P_{2j}, we form the pair of extended trees $P_1 * P_2$:

$\ldots, P_{1i-2}, P_{1i-1}, P_{1i}, P_{2j}, P_{2j+1}, P_{2j+2}, \ldots$

$\ldots, P_{2j-2}, P_{2j-1}, P_{2j}, P_{1i}, P_{1i+1}, P_{2i+2}, \ldots,$

which would form the feature set for tree kernel learning in addition to the original trees P_1 and P_2.

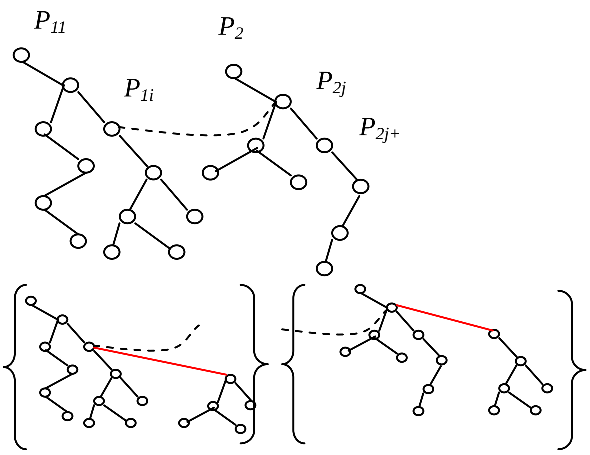


Fig. 1: An arc which connects two parse trees for two sentences in a text (on the top) and the derived set of extended trees (on the bottom).

The algorithm for building an extended tree for a set of parse trees *T* is presented below:

Input:
1) Set of parse trees *T*.
2) Set of relations *R*, which includes relations R_{ijk} between the nodes of T_i and T_j: $T_i \in T$, $T_j \in T$, $R_{ijk} \in R$. We use index *k* to range over multiple relations between the nodes of a parse tree for a pair of sentences.

Output: the exhaustive set of extended trees *E*.

Set $E = \varnothing$;
For each tree i=1:|T|
  For each relation R_{ijk}, k= 1: |R|
    Obtain T_j
    Form the pair of extended trees $T_i * T_j$;
    Verify that each of the extended trees do not have a super-tree in *E*
      If verified, add to *E*;
Return *E*.

Notice that the resultant trees are not the proper parse trees for a sentence, but nevertheless form an adequate feature space for tree kernel learning.

There are the following processing steps used in our classifier. Each paragraph of a document is subject to sentence splitting, part-of-speech tagging, dependency parsing and chunking. We also rely on additional tags to extend SVM feature space, finding similarities between trees. These additional tags include noun entities from Stanford NLP such as organization and title, and verb types from VerbNet. We then produce a graph-based representation for a document, applying anaphora and RST parser (Joty et al., 2012, 2013, 2014) for inter-sentence relations. To obtain the anaphora links, we employ coreferences from Stanford NLP (Lee et al., 2013; Recasens et al., 2013).

3 Evaluation

For the action-plan document domain, we formed a set of 940 action-plan documents from the web. We also compiled the set of meta- documents on similar engineering topics, mostly containing the same keywords. The list of documents obtained from the web is available at https://code.google.com/p/relevance-based-on-parse-trees/source/browse/src/test/resources/tree_kernel/action-plan-doc-list.csv. We split the data into 3 subsets for training/evaluation portions and cross-validation (Kohavi, 1995).

Table 1. Evaluation results.

Method	**Precision**	**Recall**	**F-measure**
*Nearest neighbor classifier (TF*IDF based)*	53.9	62	57.67+-0.62
Naive Bayesian classifier	55.3	59.7	57.42+-0.84
Tree kernel – regular parse trees	71.4	76.9	74.05+-0.55
Tree kernel SVM – extended trees for anaphora	77.8	81.4	79.56+-0.70
Tree kernel SVM – extended trees for RST	80.1	80.5	80+-1.03
Tree kernel SVM – extended trees for both anaphora and RST	**83.3**	**83.6**	**83.45+-0.78**

Table 1 shows evaluation results. Each row shows the results of the baseline classification methods, such as keyword statistics (Croft et al., 2008; Sulton and Buckley, 1998), Nearest-Neighbor classification and Naïve Bayes approach (Moore and Boyer, 1991; John and Langley, 1995).

Baseline approaches show rather low performance. The one of the tree kernel based methods improves as the sources of linguistic properties are expanded. For both domains, there is an improvement by a few percent due to the rhetoric relations compared with the baseline tree kernel SVM which employs parse trees only. For the literature documents, the role of anaphora is lower than for technical ones.

4 Discussion and Conclusions

In this study we addressed the issue of how semantic discourse features assist with solving such abstract classification problem as differentiating between natural language-object and natural meta-language. We demonstrated that the problem of such level of abstraction can nevertheless be dealt with statistical learning allowing automated feature engineering. Evaluation domain is selected so that the only differences between classes are in phrasing and discourse structures (not in keywords). We also demonstrated that both of these structures are learnable.

We draw the comparison with two following sets of linguistic features: (1) *The baseline set, parse trees for individual sentences*, and (2) *Parse trees and discourse information* and showed that the enhanced set indeed improves the classification performance for the same learning framework. One can see that the baseline text classification approaches does not perform well in the classification domain as abstract and complicated as recognizing metalanguage.

We considered the following sources of relations between words in sentences: coreferences, taxonomic relations such as sub-entity, partial case, predicates for subject etc., rhetoric structure relations, and dialogue structure. A number of NLP tasks including search relevance can be improved if search results are subject to confirmation by discourse structure plus syntactic structure generalization, when answers occur in multiple sentences. In this study we employed coreferences and rhetoric relation only to identify correlation with the occurrence of metalanguage in text. Although phrase-level analysis allows extraction of weak correlation with metalanguage in text, ascend to discourse structures makes detection of metalanguage more reliable. In our evaluation setting, using discourse improved the classification F-measure by 5.5 – 8.6% depending on a classification sub-domain.

There is a strong disattachment between modern text learning approaches and text discourse theories. Usually, learning of linguistic structures in NLP tasks is limited to keyword forms and frequencies. On the other hand, most theories of semantic discourse are not computational in nature. In this work we attempted to achieve the best of both worlds: learn complete parse tree information augmented with an adjustment of discourse theory allowing computational treatment.

In this paper, we used extended parse trees instead of regular ones, leveraging available discourse information, for text classification. This work describes one of the first applications of tree kernel to industrial scale NLP tasks. The advantage of this approach is that the manual thorough analysis of text can be avoided for complex text classification tasks where the classes are as high-level as documents vs meta-documents. The reason of the satisfactory performance of the proposed classification method is a robustness of statistical learning algorithms to noisy and inconsistent features extracted from documents.

The experimental environment, extended tree learning functionality and the evaluation framework are available at http://code.google.com/p/relevance-based-on-parse-trees.

Acknowledgements

This work was carried out by the authors within the project "Data mining based on applied ontologies and lattices of closed descriptions" supported by the Basic Research Program of the National Research University Higher School of Economics. This work has been also supported by the RFBR grant №13-07-00504/15.

References

Cumby, C. and Roth D. (2003) On Kernel Methods for Relational Learning. ICML, pp. 107-14.

Russell, S., Wefald, E, .Karnaugh, M., Karp, R., McAllester, D., Subramanian, D., Wellman, M. (1991) Principles of Metareasoning, Artificial Intelligence, pp 400--411, Morgan Kaufmann.

Collins, M., and Duffy, N. (2002) Convolution kernels for natural language. In Proceedings of NIPS, 625–32.

Mann, W. and Thompson, S. 1988. Rhetorical Structure Theory: Toward a functional theory of text organization, Text 8 (3), 243-281.

Galitsky, B. 2003. Natural Language Question Answering System: Technique of Semantic Headers. Advanced Knowledge International, Adelaide, Australia.

Galitsky, B., Usikov, D., and Kuznetsov S.O. 2013. Parse Thicket Representations for Answering Multi-sentence questions. 20th International Conference on Conceptual Structures.

Galitsky B., Ilvovsky, D., Kuznetsov, S.O. (2015) Text Integrity Assessment: Sentiment Profile vs Rhetoric Structure. CICLing-2015, Cairo, Egypt.

Wu, J., Xuan Z. and Pan, D. 2011. Enhancing text representation for classification tasks with semantic graph structures, International Journal of Innovative Computing, Information and Control (ICIC), Volume 7, Number 5(B).

Haussler, D. 1999. Convolution kernels on discrete structures. UCSB Technical report.

Kong, F. and Zhou G. 2011. Improve Tree Kernel-Based Event Pronoun Resolution with Competitive Information. Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence, 3 1814-19.

Moschitti, A. 2006. Efficient Convolution Kernels for Dependency and Constituent Syntactic Trees. 2006. In Proceedings of the 17th European Conference on Machine Learning, Berlin, Germany.

A. Severyn, A. Moschitti. 2012. Structural relationships for large-scale learning of answer re-ranking. SIGIR 2012: 741-750.

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

Mann, W., Matthiessen C. and Thompson S. 1992. Rhetorical Structure Theory and Text Analysis. Discourse Description: Diverse linguistic analyses of a fund-raising text. ed. by Mann W and Thompson S.; Amsterdam, John Benjamins. pp. 39-78.

Zhang, M.; Che, W.; Zhou, G.; Aw, A.; Tan, C.; Liu, T.; and Li, S. 2008. Semantic role labeling using a grammar-driven convolution tree kernel. IEEE transactions on audio, speech, and language processing. 16(7): 1315–29.

Lee, H., Chang, A., Peirsman, Y., Chambers, N., Surdeanu, M. and Jurafsky, D. 2013. Deterministic coreference resolution based on entity-centric, precision-ranked rules. Computational Linguistics 39(4), 885-916.

Marcu, D. 1997. From Discourse Structures to Text Summaries, in I. Mani and M.Maybury (eds) Proceedings of ACL Workshop on Intelligent Scalable Text Summarization, pp. 82–8, Madrid, Spain.

Kohavi, R. 1995. A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. International Joint Conference on Artificial Intelligence. 1137-43.

Recasens, M., de Marneffe M-C, and Potts, C. 2013. The Life and Death of Discourse Entities: Identifying Singleton Mentions. In Proceedings of NAACL.

Croft, B., Metzler, D., Strohman, T. 2009. Search Engines - Information Retrieval in Practice. Pearson Education. North America.

Salton, G. and Buckley, C. 1988. Term-weighting approaches in automatic text retrieval. Information Processing & Management 24(5): 513—23.

Moore, JS and Boyer RS. 1991. MJRTY - A Fast Majority Vote Algorithm, In R.S. Boyer (ed.), Automated Reasoning: Essays in Honor of Woody Bledsoe, Automated Reasoning Series, Kluwer Academic Publishers, Dordrecht, The Netherlands, pp. 105-17.

John G.H. and Langley, P. 1995. Estimating Continuous Distributions in Bayesian Classifiers. In Eleventh Conference on Uncertainty in Artificial Intelligence, San Mateo, 338-45.

Vapnik, V. (1995) The Nature of Statistical Learning Theory, Springer-Verlag.

Michael T. Cox and Anita Raja. (2007) Metareasoning: A manifesto.

S. Joty, G. Carenini, and R. T. Ng. 2012. A Novel Discriminative Framework for Sentence-Level Discourse Analysis. In Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, EMNLP-CoNLL'12, pages 904–915, Jeju Island, Korea. Association for Computational Linguistics.

S. Joty, G. Carenini, R. Ng, Y. Mehdad. 2013. Combining Intra- and Multi-sentential Rhetorical Parsing for Document-level Discourse Analysis. In Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, Sofia, Bulgaria.

Joty S. and Moschitti A. 2014. Discriminative Reranking of Discourse Parses Using Tree Kernels. In proceedings of EMNLP 2014.

Galitsky, B., Kuznetsov S. 2008. Learning communicative actions of conflicting human agents. J. Exp. Theor. Artif. Intell. 20(4): 277-317 .

Galitsky, B. 2012. Machine Learning of Syntactic Parse Trees for Search and Classification of Text. Engineering Applications of Artificial Intelligence. 26 (3), 1072-91