

# Going beyond sentences when applying tree kernels

Dmitry Ilvovsky

School of Applied Mathematics and Information Science  
National Research University Higher School of Economics  
Moscow, Russia

dilvovsky@hse.ru

## Abstract

We go beyond the level of individual sentences applying parse tree kernels to paragraphs. We build a set of extended trees for a paragraph of text from the individual parse trees for sentences and learn short texts such as search results and social profile postings to take advantage of additional discourse-related information. Extension is based on coreferences and rhetoric structure relations between the phrases in different sentences. We evaluate our approach, tracking relevance classification improvement for multi-sentence search task. The search problem is formulated as classification of search results into the classes of relevant and irrelevant, learning from the Bing search results. We compare performances of individual sentence kernels with the ones for extended parse trees and show that adding discourse information to learning data helps to improve classification results.

## 1 Introduction

In spite of substantial efforts to formulate a complete linking theory between syntax and semantics, it is not available yet. Hence the design of syntactic features for automated learning of syntactic structures is still an art. One of the solutions to systematically treat these syntactic features – tree kernels built over syntactic parse trees. Convolution tree kernel (Collins and Duffy, 2002) defines a feature space consisting of all subtree types of parse trees and counts the number of common subtrees as the syntactic similarity between two parse trees. They have found a number of applications in several natural language tasks, e.g. syntactic parsing re-ranking, relation extraction (Zelenko et al., 2003; Zhang et al 2006), named entity recognition (Cumby

and Roth, 2003) and Semantic Role Labeling (Moschitti, 2004), pronoun resolution (Yang et al., 2006), question classification (Zhang and Lee, 2003) and machine translation (Zhang and Li, 2009).

The kernel ability to generate large feature sets is useful 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.

Many learning algorithms, such as SVM (Vapnik, 1998) 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 objects in NLP. A number of NL tasks require computing of semantic features over paragraphs of text containing multiple sentences. Doing it in a sentence pair-wise manner is not always accurate, since it is strongly dependent on how information (phrases) is distributed through sentences.

An approach to build a kernel based on more than a single parse tree has been proposed (Severyn et.al., 2012), however without any relations between parse trees or for a different purpose than treating multi-sentence portions of text. To compensate for parsing errors (Zhang et al., 2008), a convolution kernel over packed parse forest (Severyn and Moschitti, 2012; Aioli et.al, 2007) is used to mine syntactic features from it directly. A packed forest compactly encodes exponential number of n-best parse trees, and thus containing much more rich structured features than a single parse tree. This advantage enables the forest kernel not only to be more robust against parsing errors, but also to be able to learn more reliable feature values and help to

solve the data sparseness issue that exists in the traditional tree kernel.

On the contrary, in this study we form a tree forest of sequence of sentences in a paragraph of text. Currently, kernel methods tackle individual sentences. However, in learning settings where texts include multiple sentences, structures which include paragraph-level information need to be employed. We demonstrate that in certain domains and certain cases discourse structure is essential for proper classification of texts.

## 2 Necessity to extend parse trees

We introduce a domain where a pair-wise comparison of sentences is insufficient to properly learn certain semantic features of texts. 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.

Firstly, note that keyword-based analysis does not help to separate the first three paragraphs 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.

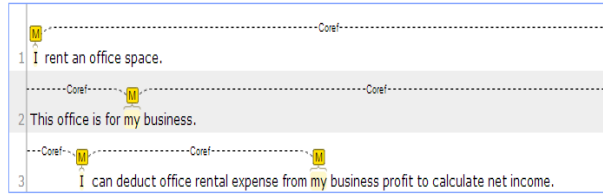
Which representations for these paragraphs of text would produce such common sub-structure between the structures of these paragraphs? We believe that extended trees, which include the first, second, and third sentence for each paragraph together can serve as a structure to differentiate the two above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

The dependency parse trees for the first text in our set and its coreferences are shown in Fig. 1. 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 used Stanford Core NLP, coreferences resolution (Lee et al., 2012) and its visualization to form Figs. 1 and 2.

Fig. 2 shows 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.

Coreference:



Basic dependencies:

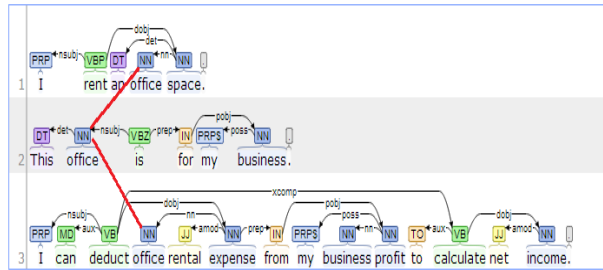


Fig.1: Coreferences and the set of dependency trees for the first text.

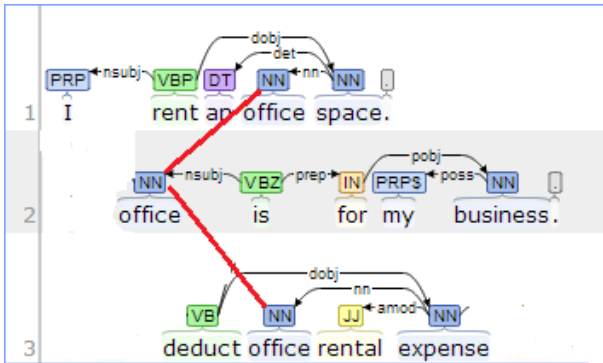


Fig. 2: Extended tree which includes 3 sentences

### 3 Building extended trees

For every arc which connects two parse trees, we derive the extension of these trees, extending branches according to the arc (Fig. 3).

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  $P_1 * P_2$ :

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

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

which would form the feature set for tree kernel learning in addition to the original trees  $P_1$  and  $P_2$ . Notice that the original order of nodes of parse trees is retained under operation "\*" (Fig. 3).

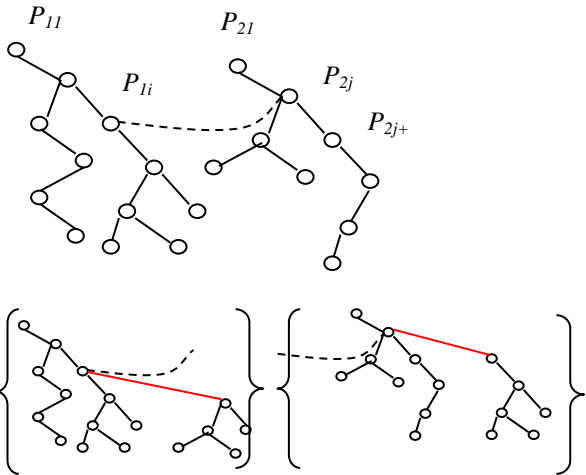


Fig. 3: An arc which connects two parse trees for two sentences in a text (on the top) and the derived set of extended trees (on the bottom).

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 parse tree for a pair of sentences.

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

Set  $E = \emptyset$ ;

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.

To obtain the inter-sentence links, we employed the following sources:

- Coreferences from Stanford NLP (Recasens et al., 2013, Lee et al., 2013).
- Rhetoric relation extractor based on the rule-based approach to finding relations between elementary discourse units (Galitsky et al., 2013). We combined manual rules with automatically learned derived from the available discourse corpus by means of syntactic generalization.

#### 4 Assessment of classification improvement

To confirm that using a set of extended parse trees for paragraphs leverages additional semantic information compared to a set of parse trees for all sentences in a paragraph, we perform an evaluation of relevance in search domain:

- As a baseline, we take all trees for sentences in paragraphs
- As an expected improvement, we take all extended trees in a paragraph.

Since a benchmarking database for answering complex multi-sentence questions is not available, we form our own dataset for product-related opinions. The question answering problem is formulated as finding information on the web, relevant to a user posting / opinion expression in a blog, forum or social network.

For the purpose of this evaluation it is not essential to provide the best possible set of answers. Instead, we are concerned with the comparison of relevance improvement by using extended parse tree, as long as the evaluation settings of question answering are identical. The details of the evaluation are given in Section 7.

#### 5 Implementation of kernel learning for extended trees

The evaluation framework described here is implemented as an OpenNLP contribution. It relies on the following systems:

- OpenNLP/Stanford NLP parser;
- Stanford NLP Coreference;
- Bing search;
- Wrapper of TK-Light kernel learner (Moschitti, 2006).

Framework includes the following components of Apache OpenNLP.similarity project:

- Rhetoric parser
- Parse thicket builder and generalizer (Galitsky et al., 2012). Not used in this evaluation.
- A number of applications based on the above component, including search (request handler for SOLR), speech recognition, content generation and others.

One of the use cases of this OpenNLP.similarity component is a Java wrapper for tree kernel algorithms implemented in C++. It allows seamless integration of tree kernel algorithms into other open source systems available in Java for search, information retrieval and machine learning. Moreover, tree kernel algorithms can be embedded into Hadoop framework in the domains where offline performance is essential. Libraries and evaluation results described in this paper are also available at <http://code.google.com/p/relevance-based-on-parse-trees> and <http://svn.apache.org/repos/asf/opennlp/sandbox/opennlp-similarity/>.

#### 6 Complexity estimation

To estimate the complexity of building extended trees, let us consider an average case with 5 sentences in each paragraph and 15 words in each sentence. We have on average 10 inter-sentence arcs, which give us up to 20 extended trees formed from two sentences, and 60 extended trees formed from 3 sentences. Hence we have to apply tree learning to up to 100 trees (of a bigger size) instead of just 5 original trees. We observe that kernel learning of extended trees has to handle at least 20 times bigger input set.

However, most of the smaller subtrees are repetitive and will be reduced in the course of dimensionality reduction.

#### 7 Evaluation

To estimate whether additional high-level semantic and discourse information contributes to classical kernel based approach, we compare two sources for trees:

- Regular parse trees
- Extended parse trees

To perform this estimation, we need a corpus including a high number of short texts similar to our example in Introduction. These texts should have high similarity (otherwise keyword approach would do well), certain discourse structure, and describe some objects (products) in a meaningful application domain. Unfortunately, to the best of our knowledge such corpus is not available. Therefore, for comparison of tree kernel performances we decided to use search results, given the query which is a short text. We rely on search engine APIs following the evaluation settings in the studies on answering complex questions (Galitsky et al., 2013).

Search results typically include texts of fairly high similarity, which is leveraged in our evaluation. To formulate classification problem on the set of texts obtained as search results, we need to form positive and negative sets. To do that, we select the first  $n$  search results as relevant (positive) and also  $n$  results towards to tail of search results lists as irrelevant (negative). In this case each search session yields an individual training (and evaluation) dataset. The same nature of such data allows averaging of precision and recall, having individual training dataset of a limited size. Hence reliability of our results is achieved not via the size of individual dataset, but instead by the increased number of search sessions. To assure an abrupt change in relevance proceeding from the head to the tail of search results lists, we use complicated queries including multiple sentences, which are not handled by modern search engines well.

The preparation of search queries (which include multiple sentences) is based on the following steps:

1. Forming the names of products and their short descriptions
2. Given (1), find a text including an extended review or opinion about this product.
3. Texts (2) cannot be used as queries as they are. To form the queries from (2), we need to extract most significant phrases from them; otherwise, search engines are confused which keywords to choose and give either duplicate, or irrelevant results. These were the longest noun and selected verb phrases from (2).

The analogous steps were conducted for Yahoo Answers data. We manually select a 100 most interesting search queries for each domain.

The training/evaluation datasets is formed from search results in the following way. We obtain a first hundred search results (or less if hundred is not available). We select 1..20 (or first 20%) of search results as a positive set, and 81..100 as a negative set. Search results 21..80 form the basis of evaluation dataset, from which we randomly select 10 texts to be classified into the classes of positive or negative. Hence we have the ratio 4:1 between the training and evaluation datasets.

To motivate our evaluation setting, we rely on the following observations. In case of searching for complex multi-sentence queries, relevance indeed drops abruptly with proceeding from the first 10-20 search results, as search evaluation results demonstrated (Galitsky et al., 2013). The order of search results in first 20% and last 20% does not affect our evaluation. Although the last 20% of search results is not really a “gold standard”, it is nevertheless a set that can be reasonably separated from the positive set. If such separation is too easy or too difficult, it would be hard to adequately evaluate the difference between regular parse trees and extended trees for text classification. Search-based approach to collect texts for evaluation of classification allows reaching maximum degree of experiment automation.

It turned out that the use of tail search results as negative set helps to leverage the high level semantic and discourse information. Negative examples, as well as positive ones, include most keywords from the queries. However, the main difference between the positive and negative search results is that the former include much more coreferences and rhetoric structures similar to the query, than the latter set. The use of the extended trees was beneficial in the cases where phrases from queries are distributed through multiple sentences in search results.

We conducted two independent experiments for each search session, classifying search result snippets and also original texts, extracted from webpages. For the snippets, we split them into sentence fragments and built extended trees for these fragments of sentences. For original texts, we extracted all sentences related to the snippet fragments and built extended trees for these sentences.

Training and classification occurs in the automated mode, and the classification assessment is

conducted by the members of research group guided by the authors. The assessors only consulted the query and answer snippets.

We used the standard parameters of tree sequence kernels from <http://disi.unitn.it/moschitti/Tree-Kernel.htm> (Moschitti, 2006). Tree kernel is applied to all tree pairs from two forests. The latest version of tree kernel learner was obtained from the author.

<i>Products</i>		<i>Basic kernels</i>	<i>Extended kernels (co-refs+RST)</i>
<b><i>Texts from the pages</i></b>	<i>Precision</i>	0,5679	0,5868
	<i>Recall</i>	0,7516	0,8458
	<i>F-measure</i>	0,6485	0,6752
<b><i>Snippets</i></b>	<i>Precision</i>	0,5625	0,6319
	<i>Recall</i>	0,7840	0,8313
	<i>F-measure</i>	0,6169	0,6695

Table 1: Evaluation results for products domain

<i>Answers</i>		<i>Basic kernels</i>	<i>Extended kernels (corefs)</i>	<i>Extended kernels (corefs+RST)</i>
<b><i>Texts from the pages</i></b>	<i>P</i>	0,5167	0,5083	0,5437
	<i>R</i>	0,7361	0,7917	0,8333
	<i>F</i>	0,6008	0,5458	0,6278
<b><i>Snippets</i></b>	<i>P</i>	0,5950	0,6264	0,6794
	<i>R</i>	0,7329	0,7492	0,7900
	<i>F</i>	0,6249	0,6429	0,7067

Table 2: Evaluation results for popular answers domain

Evaluation results show visible improvement of classification accuracy achieved by extended trees. For Yahoo Answers one can observe that coreferences only provide a slight improvement of accuracy, whereas RST added to coreferences gives a stronger improvement. Stronger increase of recall in comparison to precision can be explained by the following. It is due to the acquired capability of extended trees to match phrases from the search results distributed through multiple sentences, with questions.

## 8 Conclusions and future work

In this study we focused on how discourse information can help with text relevance tasks irrespectively of learning mechanisms. We compared two sets of linguistic features:

- The baseline, parse trees for individual sentences,

- Parse trees and discourse information,

and demonstrated that the enriched set of features indeed improves the classification accuracy, having the learning framework fixed. This improvement varies from 2 to 8 % in different domains with different structure of texts. To tackle such enriched set of linguistic features, an adjustment of tree kernel algorithm itself was not necessary.

The approach developed in this paper can also be applied to parse tree querying and manipulation problem (Levy and Galen, 2006). A system such as Tregex is an expressive and flexible way for single sentence parse tree querying and manipulation. Extending parse trees of individual sentences towards paragraph of text, the recall of a tree querying system would dramatically increase, and dependence on how phrases are distributed through sentences would decrease.

There are a few possible directions of future development. One interesting continuation of this study is to applying standard ranking mechanisms such as NDCG. We can draw the comparison between the standard and extended kernels in terms of standard Bing ranking, as well as special ranking based on syntactic similarity between the query and search results (Galitsky et al., 2013).

We also plan to generalize extended tree kernels towards graphs (DAGs) (Suzuki et al., 2003). In this case we can perform learning on Parse thickets (Galitsky et al., 2013) – the structures which are the sets of parse trees for a paragraph. It will be fruitful to compare performances of various ways of kernel computation and estimate the contribution of a particular way of paragraph representation to the quality of classification.

It is possible to apply the outlined approach to perform question answering in the case where the latter are extensive portions of paragraph-sized text and the former include multiple sentences.

Another obvious direction is applying tree kernels to classify short texts based on standard corpus data. However, a corpus of short texts, where advantages of kernel methods over alternatives would become visible, does not exist. One of our next tasks is to form such a corpus.

## Acknowledgments

We would like to thank Baidu for travel and conference support for this paper.

## References

- Cumby, C., Roth, D. 2003. *Kernel methods for relational learning*. In: ICML.
- Kim, Jung-Jae, Pezik, P. and Rebholz-Schuhmann, D. 2008. *MedEvi: Retrieving textual evidence of relations between biomedical concepts from Medline*. *Bioinformatics*. Volume 24, Issue 11 pp. 1410-1412.
- Zelenko, D., Aone, C., Richardella, A. 2003. *Kernel methods for relation extraction*. *JMLR* (2003).
- Suzuki, J., Hirao, H., Sasaki, Y and Maeda, E., *Hierarchical Directed Acyclic Graph Kernel: Methods for Structured Natural Language Data*. In Proceedings of the 41th Annual Meeting of Association for Computational Linguistics (ACL). 2003.
- Galitsky, B. *Natural Language Question Answering System: Technique of Semantic Headers*. Advanced Knowledge International, Australia (2003).
- Galitsky, B., de la Rosa, J., Dobrocsi, G. 2012. *Inferring the semantic properties of sentences by mining syntactic parse trees*. *Data & Knowledge Engineering*. Volume 81-82, November (2012) 21-45.
- Galitsky, B., Usikov, D. Kuznetsov, S. 2013. *Parse Thicket Representations for Answering Multi-sentence questions*. 20th International Conference on Conceptual Structures, ICCS 2013.
- 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 Application of AI*, <http://dx.doi.org/10.1016/j.engappai.2012.09.017>.
- Galitsky, B., Ilvovsky, D., Kuznetsov, S., Strok, F. 2013. *Improving Text Retrieval Efficiency with Pattern Structures on Parse Thickets*, Workshop "Formal Concept Analysis meets Information Retrieval" at ECIR 2013, Moscow, Russia.
- Ehrlich H.-C., Rarey M. 2011. *Maximum common subgraph isomorphism algorithms and their applications in molecular science: review*. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, 2011, vol. 1 (1), pp. 68-79.
- Yan, X., Han, J. 2002. *gSpan: Graph-Based Substructure Pattern Mining*. In: Proc. IEEE Int. Conf. on Data Mining, ICDM'02, IEEE Computer Society (2002), pp 721-724.
- Jiangning Wu, Zhaoguo Xuan and Donghua Pan, *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*.
- Moschitti, A. 2006. *Efficient Convolution Kernels for Dependency and Constituent Syntactic Trees*. In Proceedings of the 17th European Conference on Machine Learning, Berlin, Germany.
- Severyn, A., Moschitti, A. 2012. *Structural relationships for large-scale learning of answer re-ranking*. SIGIR 2012: 741-750.
- Severyn, A., Moschitti, A. 2012. *Fast Support Vector Machines for Convolution Tree Kernels*. *Data Mining Knowledge Discovery* 25: 325-357.
- Aioli, F., Da San Martino, G., Sperduti, A. and Moschitti, A. 2007. *Efficient Kernel-based Learning for Trees*, Proceeding of the IEEE Symposium on Computational Intelligence and Data Mining (CIDM), Honolulu, Hawaii.
- Punyakanok, V., Roth, D., & Yih, W. 2004. *Mapping dependencies trees: an application to question answering*. In: Proceedings of AI & Math, Florida, USA.
- Mann, William C., Christian M. I. M. Matthiessen and Sandra A. Thompson. 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, John Benjamins: 39-78.
- Sun, J., Min Zhang, Chew Lim Tan. 2011. *Tree Sequence Kernel for Natural Language*. *AAAI-25*, 2011.
- 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):1315-1329.
- Montaner, M.; Lopez, B.; de la Rosa, J. L. (June 2003). *A Taxonomy of Recommender Agents on the Internet*. *Artificial Intelligence Review* 19 (4): 285-330.
- Collins, M., and Duffy, N. 2002. *Convolution kernels for natural language*. In *Proceedings of NIPS*, 625-632, 2002.
- Heeyoung Lee, 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).
- Daniel Jurafsky, James H. Martin. 2008. *Speech and Language Processing. An Introduction to Natural Language Processing*, Computational Linguistics, and Speech Recognition.
- Robinson J.A. 1965. *A machine-oriented logic based on the resolution principle*. *Journal of the Association for Computing Machinery*, 12:23-41.



- Mill, J.S. 1843. *A system of logic, ratiocinative and inductive*. London.
- Fukunaga, K. *Introduction to statistical pattern recognition (2nd ed.)*, Academic Press Professional, Inc., San Diego, CA, 1990.
- Mitchell, T. 1997. *Machine Learning*. McGraw Hill.
- Furukawa, K. 1998. *From Deduction to Induction: Logical Perspective. The Logic Programming Paradigm*. In Apt, K.R., Marek V.W., Truszczynski, M., Warren, D.S., Eds. Springer.
- Bharat Bhasker; K. Srikumar. 2010. *Recommender Systems in E-Commerce*. CUP. ISBN 978-0-07-068067-8.
- Trias i Mansilla, A., JL de la Rosa i Esteva. 2012. *Asknext: An Agent Protocol for Social Search*. Information Sciences 190, 144–161.
- Punyakanok, V., Roth, D. and Yih, W. 2005. *The Necessity of Syntactic Parsing for Semantic Role Labeling*. IJCAI-05.
- Domingos P. and Poon, H. 2009. *Unsupervised Semantic Parsing*, In: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, Singapore: ACL.
- 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.
- Abney, S. 1991. *Parsing by Chunks, Principle-Based Parsing*, Kluwer Academic Publishers, 1991, pp. 257-278.
- Hyeran Byun, Seong-Whan Lee. 2002. *Applications of Support Vector Machines for Pattern Recognition: A Survey*. In Proceedings of the First International Workshop on Pattern Recognition with Support Vector Machines (SVM '02), Seong-Whan Lee and Alessandro Verri (Eds.). Springer-Verlag, London, UK, UK, 213-236.
- Chris Manning and Hinrich Schütze, *Foundations of Statistical Natural Language Processing*, MIT Press. Cambridge, MA: May 1999.
- Sun, J.; Zhang, M.; and Tan, C. 2010. *Exploring syntactic structural features for sub-tree alignment using bilingual tree kernels*. In *Proceedings of ACL*, 306–315.
- Kohavi, Ron. 1995. *A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection*. International Joint Conference on Artificial Intelligence IJCAI 1995.
- Kalervo Jarvelin, Jaana Kekalainen. 2002. *Cumulated gain-based evaluation of IR techniques*. ACM Transactions on Information Systems 20(4), 422–446.
- Roger Levy and Galen Andrew, Tregex and Tsurgeon: tools for querying and manipulating tree data structures. 5th International Conference on Language Resources and Evaluation (LREC 2006), 2006.
- Heeyoung Lee, Marta Recasens, Angel Chang, Mihai Surdeanu, and Dan Jurafsky. 2012. *Joint Entity and Event Coreference Resolution across Documents*. In EMNLP-CoNLL 2012.
- Marta Recasens, Marie-Catherine de Marneffe, and Christopher Potts. 2013. *The Life and Death of Discourse Entities: Identifying Singleton Mentions*. In Proceedings of NAACL 2013.