IJCLA VOL. 5, NO. 1, JAN-JUN 2014, PP. 105–116
RECEIVED 02/01/14 ACCEPTED 06/02/14 FINAL 11/06/14

Extending Tree Kernels towards Paragraphs

BORIS GALITSKY,[1,2] DMITRY ILVOVSKY,[2] AND SERGEY O. KUZNETSOV[2]

[1] *Knowledge Trail Inc., USA*
[2] *Higher School of Economics, Russia*

ABSTRACT

We extend parse tree kernels from the level of individual sentences towards the level of paragraph to build a framework for learning short texts such as search results and social profile postings. We build a set of extended trees for a paragraph of text from the individual parse trees for sentences. It is performed based on coreferences and Rhetoric Structure relations between the phrases in different sentences. Tree kernel learning is applied to extended trees to take advantage of additional discourse-related information. We evaluate our approach, tracking relevance improvement for multi-sentence search, comparing performances of individual sentence kernels with the ones for extended parse trees. The search problem is formulated as classification of search results into the classes of relevant and irrelevant, learning from the Bing search results, used as a baseline and as a training dataset.

1 INTRODUCTION

Convolution tree kernel [6] 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 [50], named entity recognition [8] and Semantic Role Labeling [53], pronoun resolution [49], question classification [52] and machine translation.

The kernel's ability to generate large feature sets is useful to model quickly new and not well-understood linguistic phenomena in learning machines. However, it is often possible to design manually features for linear kernels that produce high accuracy and low computation time, whereas the complexity of tree kernels may prevent their application in real scenarios.

Many learning algorithms, such as SVM 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 pairwise 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, however for a different purpose than treating multi-sentence portions of text. To compensate for parsing errors [51], a convolution kernel over *packed* parse forest 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 from a tree forest of sequence of sentences in a paragraph of text. In learning settings where texts include multiple sentences, structures that 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 FROM REGULAR TO EXTENDED TREES

For every arc that 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.

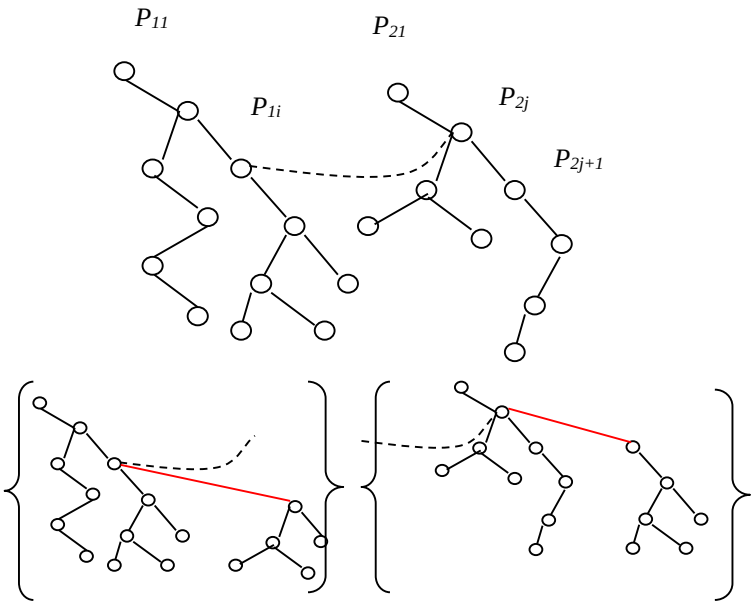

Fig. 1. An arc that connects two parse trees for two sentences in a text (top) and the derived set of extended trees (bottom).

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 $P_1 * P_2$:

…, P_{1i-2}, P_{1i-1}, P_{1i}, P_{2j}, P_{2j+1}, P_{2j+2}, …,
…, P_{2j-2}, P_{2j-1}, P_{2j}, P_{1i}, P_{1i+1}, P_{2i+2}, …,

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

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 $E = \varnothing$;
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.

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

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

1. Co-reference tools from Stanford NLP [41, 26].
2. Rhetoric relation extractor based on the rule-based approach to finding relations between elementary discourse units [15, 16]. We combined manual rules with automatically learned rules derived from the available discourse corpus by means of syntactic generalization.

3 Implementation of Paragraph Learning

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 kernel learner [36].

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. Code and libraries described here 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.

4 Complexity Estimation

To estimate the complexity of building extended trees, let us consider an average case with five 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 three 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. In addition, in an industrial search application where phrases are stored in an inverse index, the generalization operation can be completed in constant time, irrespectively of the size of index [29]. In case of map-reduce implementation of generalization operation, for example, using Cascading framework, the time complexity becomes constant with the size of candidate search results to be re-ranked [9].

5 Evaluation of Multi-sentence Classification in Search Domain

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. We apply the same type of tree kernel learning for a paragraph, obtaining parse trees by following two ways:

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

We then compare the classification results as obtained by tree kernel algorithm, applied to the two above sources. We select a search domain that allows us an unlimited set of paragraph-level text initiating search. However, tree kernels are used to solve search relevance problem as classifying candidate answers to be relevant or not. We use Bing search engine API for all web mining and search baseline tasks.

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. We generate a set of queries as short paragraphs of text and run Bing web search engine API to find a candidate set of answers and form a training set.

The classification problem is formulated as classifying a set of search results into the classes of relevant and irrelevant. The respective training dataset is formed from the set of highly ranked answers (as a positive, relevant set) and the set of answers with lower rank (as a negative, irrelevant set). Some randomly selected other candidates are classified, given this training dataset. For each candidate search result, we use its snippet as obtained by Bing and the respective portion of text extracted from the webpage. This experiment is based on the suggestion that top (bottom) Bing results are somehow relevant (irrelevant) to the initial query despite that they can be ordered in a wrong way.

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 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 [15, 16]. 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 we hard to evaluate adequately 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. Use of 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 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 for 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 use the standard parameters of tree sequence kernels from http://disi.unitn.it/moschitti/Tree-Kernel.htm [36]. The latest version of tree kernel learner was obtained from the author.

Table 1. Evaluation results for products domain

Products		Basic kernels [36]	Extended kernels
Text from the page	Precision	0.568	0.587
	Recall	0.752	0.846
	F-measure	0.649	0.675
Snippets	Precision	0.563	0.632
	Recall	0.784	0.831
	F-measure	0.617	0.670

Table 2. Evaluation results for popular answers domain

Answers		Basic kernels [36]	Extended kernels
Text from the page	Precision	0.517	0.544
	Recall	0.736	0.833
	F-measure	0.601	0.628
Snippets	Precision	0.595	0.679
	Recall	0.733	0.790
	F-measure	0.625	0.707

Evaluation results show visible improvement of classification accuracy achieved by extended trees. 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.

6 CONCLUSIONS

In this study 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.

Traditionally, machine learning of linguistic structures is limited to keyword forms and frequencies. At the same time, most theories of discourse are not computational, they model a particular set of relations between consecutive states. 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.

The experimental environment, multi-sentence queries and the evaluation framework is available at http://code.google.com/p/relevance-based-on-parse-trees.

REFERENCES

1. Abney, S.: Parsing by Chunks, Principle-Based Parsing, Kluwer Academic Publishers (1991) 257–278
2. Bhasker, Bharat, K. Srikumar: Recommender systems in e-Commerce. CUP (2010)
3. Bron, Coen; Kerbosch, Joep: Algorithm 457: finding all cliques of an undirected graph, Commun. ACM (ACM) **16**(9) (1973) 575–577
4. Byun, Hyeran, Seong-Whan Lee: 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, Springer, London, UK, (2002) 213–236
5. Cascading. en.wikipedia.org/wiki/Cascading; www.cascading.org/2013
6. Collins, M., and Duffy, N.: Convolution kernels for natural language. In: Proceedings of NIPS (2002) 625–632
7. Conte, D., P. Foggia, C. Sansone, and M. Vento.: Thirty years of graph matching in pattern recognition. International Journal of Pattern Recognition and Artificial Intelligence **18**(3) (2004) 265–298
8. Cumby, C., Roth, D.: Kernel methods for relational learning. In: Proceedings of ICML 2003 (2003)
9. Dean, J.: Challenges in Building Large-Scale Information Retrieval Systems. research.google.com/people/jeff/WSDM09-keynote.pdf (2009)
10. Domingos, P. and Poon, H.: Unsupervised semantic parsing. In: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, Singapore, ACL (2009)
11. Ehrlich H.-C., Rarey M.: Maximum common subgraph isomorphism algorithms and their applications in molecular science: review. Wiley Interdisciplinary Reviews: Computational Molecular Science **1**(1) (2011) 68–79
12. Fukunaga, K.: Introduction to statistical pattern recognition (2nd ed.), Academic Press Professional, Inc., San Diego, CA (1990)
13. Furukawa, K.: From deduction to induction: Logical perspective. The Logic Programming Paradigm. Springer (1998)
14. Galitsky, B.: Natural language question answering system: Technique of semantic headers. In: Advanced Knowledge International, Australia (2003)
15. Galitsky, B., Daniel Usikov, Sergei O. Kuznetsov: Parse thicket representations for answering multi-sentence questions. In: 20th International Conference on Conceptual Structures, ICCS 2013 (2013)
16. Galitsky, B., Ilvovsky, D. Kuznetsov, S., Strok, F.: Improving text retrieval efficiency with pattern structures on parse thickets. In: Workshop on Formal Concept Analysis Meets Information Retrieval at ECIR 2013, Moscow, Russia (2013)
17. Galitsky, B., Josep Lluis de la Rosa, Gábor Dobrocsi: Inferring the semantic properties of sentences by mining syntactic parse trees. Data and Knowledge Engineering **81–82** (2012) 21–45

18. Galitsky, B., Kuznetsov S, Learning communicative actions of conflicting human agents. J. Exp. Theor. Artificial Intelligence **20**(4) (2008) 277–317
19. Galitsky, B., Machine learning of syntactic parse trees for search and classification of text. Engineering Application of Artificial Intelligence **26**(3) (2012) 1072–1091
20. Haussler, D. 1999. Convolution kernels on discrete structures.
21. Jarvelin, Kalervo, Jaana Kekalainen: Cumulated gain-based evaluation of IR techniques. ACM Transactions on Information Systems **20**(4) (2002) 422–446
22. Jurafsky, D., Martin, J. Speech and language processing. An introduction to natural language processing, computational linguistics, and speech recognition (2008)
23. Kann, V.: On the approximability of the maximum common subgraph problem. In: (STACS '92), Springer, London, UK (1992) 377–388
24. Kim, Jung-Jae, Piotr Pezik and Dietrich Rebholz-Schuhmann. MedEvi: Retrieving textual evidence of relations between biomedical concepts from Medline. Bioinformatics **24**(11) (2008) 1410–1412.
25. Kohavi, Ron: A study of cross-validation and bootstrap for accuracy estimation and model selection. In: International Joint Conference on Artificial Intelligence IJCAI 1995 (1995)
26. Lee, Heeyoung, Angel Chang, Yves Peirsman, Nathanael Chambers, Mihai Surdeanu and Dan Jurafsky: Deterministic coreference resolution based on entity-centric, precision-ranked rules. Computational Linguistics **39**(4), 2013.
27. Lee, Heeyoung, Marta Recasens, Angel Chang, Mihai Surdeanu, and Dan Jurafsky: Joint Entity and Event Coreference Resolution across Documents. In: Proceedings of EMNLP-CoNLL (2012)
28. Levy, Roger and Galen Andrew: Tregex and Tsurgeon: tools for querying and manipulating tree data structures. In: Proceedings of the 5th International Conference on Language Resources and Evaluation, LREC 2006 (2006)
29. Lin, J., Chris Dyer: Data-intensive text processing with MapReduce. Morgan & Claypool Publishers (2010)
30. Mann, William C., Christian M. I. M. Matthiessen and Sandra A. Thompson: Rhetorical structure theory and text analysis. In: Discourse Description: Diverse linguistic analyses of a fund-raising text. John Benjamins (1992) 39–78.
31. Manning, Chris and Hinrich Schütze: Foundations of Statistical Natural Language Processing, MIT Press. Cambridge, MA (1999)
32. Marcu, D.: From discourse structures to text summaries. In: Proceedings of ACL Workshop on Intelligent Scalable Text Summarization, Madrid, Spain (1997) 82–88
33. Mill, J.S.: A system of logic, ratiocinative and inductive. London (1843)
34. Mitchell, T.: Machine Learning. McGraw Hill (1997)
35. Montaner, M.; Lopez, B.; de la Rosa, J. L.: A taxonomy of recommender agents on the Internet. Artificial Intelligence Review **19**(4) (2003) 285–330

36. Moschitti, A.: Efficient convolution kernels for dependency and constituent syntactic trees. In: Proceedings of the 17th European Conference on Machine Learning, Berlin, Germany (2006)
37. Moschitti, Alessandro, Daniele Pighin, and Roberto Basili: Tree kernels for semantic role labeling. Computational Linguistics **34**(2) (2008) 193–224
38. Polovina S., John Heaton: An Introduction to conceptual graphs. AI Expert (1992) 36–43
39. Punyakanok, V., Roth, D., Yih, W.: Mapping dependencies trees: an application to question answering. In: Proceedings of AI & Math, Florida, USA (2004)
40. Punyakanok, V., Roth, D., Yih, W.: The Necessity of Syntactic Parsing for Semantic Role Labeling. In: Proceedings of IJCAI-05 (2005)
41. Recasens, Marta, Marie-Catherine de Marneffe, and Christopher Potts: The life and death of discourse entities: Identifying singleton mentions. In: Proceedings of NAACL 2013 (2013)
42. Robinson J.A.: A machine-oriented logic based on the resolution principle. Journal of the Association for Computing Machinery **12** (1965) 23–41
43. Sun, J., Min Zhang, Chew Lim Tan: Tree Sequence Kernel for Natural Language. AAAI-25 (2011)
44. Sun, J.; Zhang, M.; and Tan, C.: Exploring syntactic structural features for sub-tree alignment using bilingual tree kernels. In: Proceedings of ACL (2010) 306–315
45. Trias i Mansilla, A., J.L. de la Rosa i Esteva : Asknext: An agent protocol for social search. Information Sciences **190** (2012) 144–161
46. Vismara, Philippe, Benoît, Valery: Finding Maximum Common Connected Subgraphs Using Clique Detection or Constraint Satisfaction Algorithms. Modelling, Computation and Optimization in Information Systems and Management Sciences, Springer (2008)
47. Wu, Jiangning, Zhaoguo Xuan, Donghua Pan: Enhancing text representation for classification tasks with semantic graph structures. International Journal of Innovative Computing, Information and Control **7**(5(B)) (2011) 2689–2698
48. Yan, X., Han, J.: gSpan: Graph-based substructure pattern mining. In: Proceedings of the IEEE International Conference on Data Mining, ICDM'02, IEEE Computer Society (2002) 721–724
49. Yang X. F., Su J., Chew C. L.: Kernel-based pronoun resolution with structured syntactic knowledge. In: COLING-ACL'2006. (2006)
50. Zelenko, D., Aone, C., Richardella, A.: Kernel methods for relation extraction. Journal of Machine Learning Research **3** (2003) 1083–1106
51. Zhang M., Zhang H., Li H., Convolution Kernel over Packed Parse Forest. In: Proceedings of ACL-2010 (2010)
52. Zhang, Dell, Wee Sun Lee.: Question classification using Support Vector Machines. In: Proceedings of the 26th ACM SIGIR (2003) 26–32
53. Zhang, M., Che, W., Zhou, G., Aw, A., Tan, C., Liu, T., Li, S.: Semantic role labeling using a grammar-driven convolution tree kernel. IEEE Transactions on Audio, Speech, and Language Processing **16**(7) (2008) 1315–1329

BORIS GALITSKY
KNOWLEDGE TRAIL INC.
SAN JOSE, CA, USA
AND
NATIONAL RESEARCH UNIVERSITY HIGHER SCHOOL OF ECONOMICS,
KOCHNOVSKI PR. 3, MOSCOW, 125319, RUSSIA
E-MAIL: <BGALITSKY@HOTMAIL.COM>

DMITRY ILVOVSKY
NATIONAL RESEARCH UNIVERSITY HIGHER SCHOOL OF ECONOMICS,
KOCHNOVSKI PR. 3, MOSCOW, 125319, RUSSIA
E-MAIL: <DILVOVSKY@HSE.RU>

SERGEY O. KUZNETSOV
NATIONAL RESEARCH UNIVERSITY HIGHER SCHOOL OF ECONOMICS,
KOCHNOVSKI PR. 3, MOSCOW, 125319, RUSSIA
E-MAIL: < SKUZNETSOV@HSE.RU>