Journal of Computer Science 4 (5): 375-381, 2008
ISSN 1549-3636

CTSS: A Tool for Efficient Information Extraction with Soft Matching Rules for Text Mining

[1]A. Christy and [2]P. Thambidurai
[1]Sathyabama University, Jeppiaar Nagar, Chennai-119, India
[2]Perunthalaivar Kamarajar Institute of Technology, Karaikal, India

Abstract: The abundance of information available digitally in modern world had made a demand for structured information. The problem of text mining which dealt with discovering useful information from unstructured text had attracted the attention of researchers. The role of Information Extraction (IE) software was to identify relevant information from texts, extracting information from a variety of sources and aggregating it to create a single view. Information extraction systems depended on particular corpora and were poor in recall values. Therefore, developing the system as domain-independent as well as improving the recall was an important challenge for IE. In this research, the authors proposed a domain-independent algorithm for information extraction, called SOFTRULEMINING for extracting the aim, methodology and conclusion from technical abstracts. The algorithm was implemented by combining trigram model with softmatching rules. A tool CTSS was constructed using SOFTRULEMINING and was tested with technical abstracts of www.computer.org and www.ansinet.org and found that the tool had improved its recall value and therefore the precision value in comparison with other search engines.

Key words: Parsing, trigram model, soft matching, information extraction, recall, precision

INTRODUCTION

The specific notion of Information extraction has received wide attention in last decade (1990s) through the series of Message Understanding Conferences, founded by US defense research group DARPA. Researchers from NLP and IE have used common evaluations to accelerate their research progress, through these conferences. They have compared different systems to give a certain transparency to the field.

Previous studies have shown that bag of words, natural language processing techniques which may utilize rule-based grammars, part-of-speech taggers and parsers , development of templates, Learning methods, Hidden markov models, Bayesian networks, Data compression, Machine learning, etc as some of the techniques adopted in IE[7,11,21]. Bag of words is the traditional method used for extracting information like sentiment (Casey Whitelaw, Navendu Garg and Shlomo Argamon)[11], Library books categorization[32], topic ontology[7], Feature Generation for Text Categorization Using World Knowledge [22].

Text: Statistical hidden state sequence models, such as Hidden Markov Models (HMMs)[24], Conditional Markov Models (CMMs) and Conditional Random Fields (CRFs)[30] are a prominent recent approach to information extraction tasks. Some of the other systems existing for IE is extracting information on interacting proteins from biomedical text using manually developed patterns[21], extracting the names of organizations and their headquarters by generating patterns and extracting tuples from plain-text documents (Snowball system), a genre-based extraction patterns using natural language processing techniques for extracting the rhetoric information contained in technical abstracts[29], extracting a database from postings to the USENET newsgroup, Austin.jobs, etc using predefined templates[31], etc. By discovering predictive relationships between different pieces of extracted data, data mining algorithms can be used to improve the accuracy of information extraction. The recall value of an IE system is significantly lower than its precision; such predictive relationships can be productively used to improve recall by suggesting additional information to extract.

System Architecture: The objective of the system is to extract the aim, methodology and conclusion specified by authors in technical abstracts. The general architecture of a text mining system is depicted in

Corresponding Author: A. Christy, Sathyabama University, Jeppiaar Nagar, Chennai-119, India

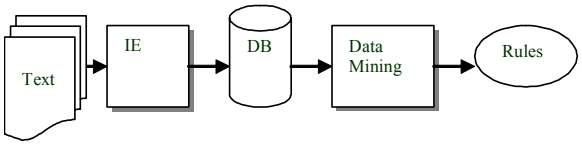

Fig. 1: General Architecture of IE systems

Fig. 1. The system deals with extracting information from multiple documents, stored in database and using data mining techniques to extract knowledge in the form of rules.

By discovering predictive relationships between different pieces of extracted data, data mining algorithms can be used to improve the accuracy of information extraction. Knowledge Discovery in Databases benefits IE by discovering rules that support predictions that can improve the accuracy of subsequent IE.

Parsing: To extract the rules, the IE task takes the set of tagged documents and produces a template representation for every document. This can be easily converted into rule-like form. For this purpose, a set of domain-independent extraction patterns are written so that we could match them against the input documents. Each extraction pattern constructs an output representation that involves two levels of linguistic knowledge: the rhetorical information expressed in the abstract and the semantic information contained in it, which we later convert into a predicate-like form. The left-hand expression states the pattern to be identified and the right hand side (following the colon) states the corresponding semantic action to be produced.

The process starts with the splitting of a given sentence into various tokens (words), from which the stop words, such as the, a, an, it, etc. are removed, as they contribute no meaning for recognition of key terms used for IE. The Morphological and lexical processing concerns how words are constructed from more basic meaning units called morphemes. A morpheme is the primitive unit of meaning in a language (the meaning of the word proposed is derivable from the meaning of the verb propose the stem word) and the inclusion of suffixes may transforms a verb into adverb. The morphological processing deals with the identification of stem word, which is a verb. Syntactic Analysis concerns how words can be put together to form correct sentences and determines what structural role each word plays in the sentence and what phrases are subparts of what other phrases. Domain analysis includes the general knowledge about the structure of the world that language users must have in order to, for example, maintain efficient knowledge discovery. The Verb Phrase (VP) is decomposed into two elements: the predicate action and the sequence of terms that represent its argument:

- Generalize (error, diffusion, produce, FM, halftone..) → Where generalize is the predicate action
- Error, diffusion,… as its argument.

The documents are parsed and the type of each and every word is analyzed. The Trigram set is used to extract the essential features and it is expressed in a tuple form like (previous token, current token, next token):

- Current Token: This is the token in its full form, as it occurs in the text. Verb is always considered as the current token
- Previous Token: This is the token to the immediate left of the current token or a special marker, if the current token is first in the sentence
- Next Token: This is the token to the immediate right of the current token or a special marker if the current token is last in the sentence

If the Current token (in the form of verb) retrieves is (be), the model retrieve the next verb as the keyword. If the sentence is in the active form, the keywords followed by the verb are retrieved and if it is in the passive form, the entire sentence from the beginning will be extracted. 15 rules which satisfy the Trigram model were written.

Softmatching rules: The IE system in this work is extracted using trigram model and rules are constructed using patterns which need not strictly adhere to the procedure. The Fig. 2 shows a sample of softmatching rules, those are introduced.

The rules are softmatching rules, as these are some frequently occurring terms which best fits the templates. Introduction of these softmatching rules have shown the improvement over the precision value, so as the recall. The algorithm SOFTRULEMINING is implemented for Information extraction using softmatching rules and is depicted in Fig. 3.

As Information extraction systems are domain specific, machine learning plays a vital role in classification and prediction. During the learning Process of machine learning, a sample of the database is used to train the system to properly perform the desired task. The quality of the training data determines how

Propose(we propose, I propose, proposes, proposed)
∈Trigram model ∈ verb ➔ aim

Method(principle, algorithm, concept, technique, work, challenge)
∈ trigram ∈ verb ➔ method

Conclude (experiment, result, test, tested, train, analytical)
∈trigram ∈ verb ➔ conclusion

Fig. 2: Sample soft matching rules

Input:

D_{train} = set of documents

Output:
T = set of new patterns for D_{test}
R = set of rules
S = set of soft rules

Function Softrulemining(D)
Build an IE rule base, RB_{IE}
(by applying BWI to D_{train})
Build a soft association rule base, RB
(by applying SOFTRULEMINING to D_{train} with parameters T_{sim} and predefined Templates)

For each unlabelled document D_{test}(i) Do
Extract slot fillers from D_{test}(i) using RB_{IE}
Let T(i) = set of extracted patterns of D_{test}
Until no change obtained on T(i) Do
For each S do
If S fires on T(i)
Extract the pattern in D_{test}(i)
Else
For each R ∈RB do
If R fires on T(i)
Extract the patterns in D_{test}(i)

For each matching pattern Y' in D_{test}(i) do
Tf = tf.log (N/df)
(with Tf < T_{sim})
if tf < T_{sim}
add Y' to T(i)
Let T = (T(1), T(2) ...T(n))
Return T.

Fig. 3: Softrulemining algorithm

well the program learns. The documents are trained with a bag of words and in order to normalize theKeywords, the inverse document frequency is used in which each document can be represented as a term vector of the form $\bar{a} = (a_1, a_2, \ldots a_n)$.

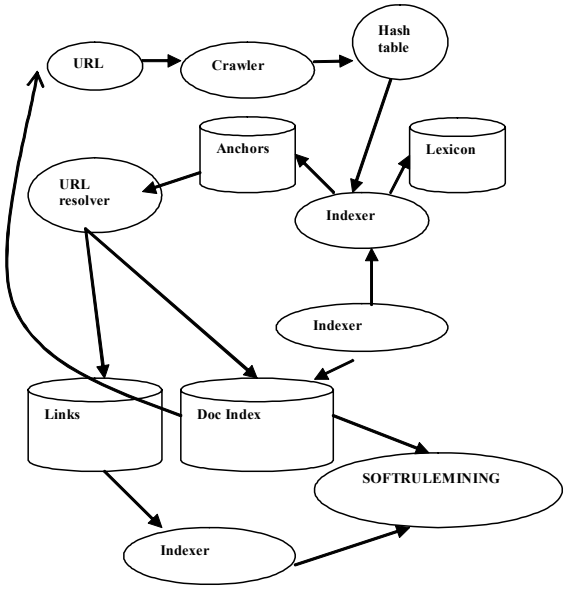

Fig. 4: Architecture of CTSS

Each term a_i has a weight w_i associated with it and w_i denotes the normalized frequency of word in the vector space , where $w_i = tf_i$. idf_i where tf_i is the term frequency of a_i, idf_i is inverse document frequency denoted as log (N/DF) where N is the total number of documents and DF is the number of documents in which a term has appeared in a text collection.

Architecture OF CTSS - information extraction tool: CTSS is a tool that is developed using Java for extracting information different URLs. This tool is implemented using the SOFTRULEMINING algorithm as the algorithm has shown better recall value than the trigram model that is adopted. Figure 4 shows the architecture of CTSS.

Given the URL as input, the web crawler fetches the pages from the links present. The system searches with the given set of patterns and if matches, it indexes the selected strings and store it in a hash table. Every web page has an associated ID number called a docID, which is assigned whenever a new URL is parsed out of a webpage. The indexing function is performed by a indexer and a sorter. The indexer performs a number of functions. It reads the hash table contents and records the word and its corresponding position in the document.

Another important function performed by the index is, it parses out all the links on web pages and stores the extracted information about them in an anchors file. This file contains enough information to determine, where each link points from and to and the text of the

- Enter the URL
- Seek to the start of the page and seek for the patterns
- Scan through the doclist and index the documents based on the frequency of occurrence of patterns
- Extract the relevant information and store them in a hash table
- If at the end of any doclist, go to step 3. Sort the documents based on relevancy and return the top k,

Fig. 5: Algorithm CTSS

link. The URL resolver reads the anchors file and converts relative URLs into absolute URLs and in turn into docIDs. It puts the anchor text into the forward index, associated with the docID that the anchor points to. It generates a database of links, which are pairs of docIDs. The links database is used to compute, indexed all documents.

The searcher is run by the webserver and uses the lexicon given by the user and the indexer to extract the information. The algorithm for the proposed approach is explained in Fig. 5.

RESULTS AND DISCUSSION

Discovered knowledge is only useful and informative if it is accurate. It is important to measure the discovered knowledge on independent test data. For the dataset, 200 abstracts were collected from www.computer.org containing 2 data sets related to information retrieval and image processing and manually annotated with correct extraction patterns. Inorder to construct the patterns classification algorithms C4.8, Random tree, Random forest, Decision tree, Decision stump were used with 10-folds cross validation. Genetic algorithms with crossover probability 0.99 and mutation level 0.01 is performed and it is found that the genetic algorithm producing better recall value compared to other classification methods. The data trained using genetic algorithm is then used for the purpose of constructing patterns.

Patterns are constructed using the tokens trained using genetic algorithm and SOFTRULEMINING is then used for information extraction. The results obtained using SOFTRULEMINING is compared with results of HMM model and Hardmatchingrules. The results are depicted in Table 1.

The patterns, which are constructed are verified using training data and tested using different domains like www.computer.org and www.ansinet.org. Three-fourth of the technical magazines from www.computer.org are checked using the proposed algorithm and it is found that the system has improved its recall value after the implementation of softmatching rules.

Table 1: Experimental Results of IE using softmatching rules

Technique	Category	Domain	Precision	Recall
	Aim	Domain 1	1	0.88
Trigram model with SOFT matching rules		Domain 2	0.98	0.86
	Methodology	Domain 1	1	0.84
		Domain 2	1	0.78
	Conclusion	Domain 1	0.94	0.87
		Domain 2	0.96	0.8
Trigram model with Hardmatching rules	Aim	Domain 1	0.9	0.83
		Domain 2	0.76	0.71
	Methodology	Domain 1	0.88	0.58
		Domain 2	0.84	0.69
	Conclusion	Domain 1	0.84	0.64
		Domain 2	0.81	0.72
HMM models	Aim	Domain 1	0.9	0.68
		Domain 2	0.9	0.82
	Methodology	Domain 1	0.8	0.64
		Domain 2	0.62	0.58
	Conclusion	Domain 1	0.82	0.71
		Domain 2	0.78	0.71

Table 2: Comparison between CTSS and Google

	Run time (Sec)	
	Google	CTSS
http://feeds.pheedo.com/ieee_intelligent_systems	33.14	32
http://feeds.pheedo.com/ieee_multimedia	48.12	37
http://feeds.pheedo.com/ieee_software	36.12	34
http://feeds.pheedo.com/it_professional	95.14	93
http://feeds.pheedo.com/ieee_computer_graphics_and_applications	80.13	78

The rules for identifying the occurrence of the current token preceded and followed by the proper order specified and finding the threshold (inverse document frequency, between 0 and 1). If the rules satisfy the condition, they are added to the rule, else it is pruned. For each rule extracted, see whether the training set of data matches the current token, if it matches the rules are extracted and stored in the structured format.

The tool is run on a P-IV system and time for extraction using google search engine and SOFTRULEMINING are studied and the proposed system has shown better recall value and saves time as compared to google search engine for extracting the specified information as shown in Table 2. Since the Google search engine fetches the relevant documents, scanning through the documents and extracting the key information is time consuming, whereas in CTSS, the webpages are directly scanned and indexed which saves time.

Evaluation: After designing a set of probabilities and an algorithm for some particular application, it is necessary to evaluate the efficiency of the algorithm. The general method for doing this is to divide the corpus into two parts: the training set and the test set. A test set consists of 10-20%of the total data. Running the algorithm on the training set is considered a reliable method of evaluation. A more thorough method of testing is called cross-validation, which involves training on the remainder of the corpus and then evaluating on the new test set.

CONCLUSION

Since the success of any machine learning algorithm depends on the type of features selected, 120 patterns were written using softmatching rules, which have improved the recall value of the information extraction system. The following are some of the findings of the system:

- In specifying the aim and conclusion authors have used only a frequent set of tokens in different domains than for specifying the methodology. More training is needed for identifying tokens for methodology
- The system is tested with different websites having technical abstracts and the introduction of softmatching rules have shown good performance over the existing methods. Therefore the proposed system can be considered as a domain-independent system
- The algorithm SOFTRULEMINING has been proposed and it has shown 84% recall value as against the other methods which have shown recall value of 70% and less
- The previous technique has dealt with a single domain as well as with manually collected documents, whereas in the proposed system, the algorithm is tested with live data from www.computer.org and www.ansinet.org. The recall value is efficient than google search engine
- The construction of patterns needed efficient learning algorithms. The system tokens are classified and trained using classification techniques like C4.8, Random tree , random forest , Decision trees, Decision stump at ten folds cross validation. Similarly classification is done using genetic algorithm at various crossover probabilities like 0.6, 0.7, 0.8 0.99 and mutation level 0.01 in which the crossover level 0.99 have shown a good recall value compared to the other methods. Therefore Genetic algorithm is used for the purpose of learning
- The SOFTRULEMINING is implemented as a tool called CTSS, which fetches abstracts from given URLs and extracts and store the information in the form of database. The proposed tool CTSS is found to show better recall value than the results obtained after extracting information through google search engine

REFERENCES

1. Aidan F. and K. Nicholas, 2006. Active learning selection strategies for information extraction. In: Proceeding of the 11th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2006, April 3-7, Trento, Italy, pp: 18-25.
2. Aidan F. and K. Nicholas, 2004. Multi-level boundary classification for information extraction, smart media institute. In: Proceeding of European Conference on Machine Learning (ECML 2004), Springer, Sep. 20-24, Pisa, 3201: 111-122.
3. Appelt and Israel, 1999. Tutorial notes of the international joint conference on artificial intelligence tutorial on information extraction technology. In: Proceeding of the 16th International Joint Conference on Artificial Intelligence (IJCAI-99), Morgan Kaufmann Publication, Stockholm, Sweden, August 1999, pp: 65-80.
4. Ayuso, D., S. Boisen, H. Fox, H. Gish, R. Ingria and R. Weischedel, 1992. BBN: Description of the PLUM system as used for MUC-4. In: Proceeding of Conference on Message Understanding (MUC-4), June 16-18, McLean, Virginia pp: 169-176. DOI: 10.3115/1072064.1072091.
5. Baumgartner, R., S. Flesca and G. Gottlob, 2001. Visual web information extraction with lixto. In: Proceeding of the 27th International Conference on Very Large Data Bases, 2001, Morgan Kaufmann Publishers Inc. San Francisco, CA, USA. ISBN: 1-55860-804-4, pp: 119-128.
6. Beale, S., Nirenburg, S. and K. Mahesh, 1995. Semantic analysis in the mikrokosmos machine translation project. In: Proceeding of the 2nd Symposium on Natural Language Processing -95, Bangkok, Thailand, pp: 297-307.
7. Blaˇz Fortuna, Dunja Mladeniˇc and Marko Grobelnik, 2005. Semi-automatic construction of topic ontology. In: Proceeding of ECML/PKDD Workshop KDO 2005, Portugal., pp: 121-131. doi: 10.1007/11908678.

8. Bloehdorn, S., P. Cimiano and A. Hotho, 2005. Learning ontologies to improve text clustering and classification. In: From Data and Information analysis to knowledge engineering. In: Proceeding of the 29th Annual Conference of the German Classification Society (GfKl 2005), Magdeburg, Germany, March 9-11, volume 30 of Studies in Classification, Data Analysis, and Knowledge Organization, pp : 334-341.
9. Califf, M.E. and R.J. Mooney, 1999. Relational learning of pattern-match rules for Information extraction. In: AAAI 99/IAAI 99: In Proceeding of the 16th National Conference on Artificial Intelligence and the 11th Innovative Applications of Artificial Intelligence Conference Innovative Applications of Artificial Intelligence, American Association for Artificial Intelligence Menlo Park, CA, USA, pp: 328-334. http://acl.ldc.upenn.edu/W/W97/W97-1002.pdf.
10. Caraballo, S.A., 1999. Automatic construction of a hypernym-labeled noun hierarchy from text. In: Proceeding of the 37th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics Morristown, NJ, USA, pp: 120-126. ISBN: 1-55860-609-3, DOI: 10.3115/1034678.1034705.
11. Casey, W., G. Navendu and A. Shlomo, 2005. Using appraisal taxonomies for sentiment analysis. In Proceeding of CIKM'05, October 31-November 5, Bremen, Germany.
12. Ciravegna, F., 2001. Adaptive information extraction from text by rule induction and generalization. In: Proceeding of the 17th International Joint Conference on Artificial Intelligence (IJCAI) August 4-10, Morgan Kaufmann Publication, Seattle, pp: 1251-1256. http://www.dcs.shef.ac.uk/~fabio/paperi/IJCAI01.pdf.
13. Crescenzi, V., G. Mecca and P. Merialdo, 2001. RoadRunner: Towards automatic data extraction from large web sites. In: Proc. of the Conference on Very Large Databases (VLDB'01), Sep. 11-14, Morgan Kaufmann Publishers Inc. San Francisco, CA, USA, pp: 109-118. http://blondie.cs.byu.edu/CS652/crescenzi01roadrunner.pdf.
14. Crescenzi, V., G. Mecca and P. Merialdo, 2001. Automatic web information extraction in the road runner system. In Proceeding of International Workshop on Data Semantics in Web Information Systems (DASWIS-2001) in conjunction with 20th International Conference on Conceptual Modeling (ER 2001), Nov. 29-30, Springer-Verlag London, UK, pp: 264-277. ISBN: 3-540-44122-0.
15. Cunningham, H., D. Maynard, K. Bontcheva and V. Tablan, 2002. GATE: A framework and graphical development environment for robust NLP tools and applications. In: Proceeding of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL'02), Association for Computational Linguistics Morristown, NJ, USA, 7 July. Philadelphia, pp: 54-62.
16. Dario, B., C. Emanuele and Vittorio, 2003. Zipping out relevant information. Comput. Sci. Eng. 5: 80-85. doi: 10.1109/MCISE. 2003.1166556.
17. Dash, M. and H. Liu, 1997. Feature selection for classification. Intel. Data Anal., 1: 131-156. doi: 10.1016/S1088-467X(97)00008-5.
18. Diego, M. and S. Rolf, R. Fabio, D. Jmaes and H. Michael, 2003. ExtrAns: Extracting answers from Technical texts. IEEE Intel. Syst., 18: 12-17. doi: 10.1109/ MIS.2003.1217623.
19. Doorenbos, R.B., O. Etzioni and D.S. Weld, 1997. A scalable comparison-shopping agent for the world-wide web. In: Proceedings of the First International Conference on Autonomous Agents (Agents'97), Feb. 5-8, Marina Del Rey, ACM Press, CA, USA, pp: 39-48.
20. Elias, F.C., M. Elena, D. Irene, R. Jose and M. Richardo, 2005. Introducing a family of linear measures for feature selection in text categorization. IEEE Trans. Knowledge Data Eng., 17: 1223-1232. doi: 10.1109/TKDE.2005.149.
21. Eugene, A. and Luis Gravano, 2000. Extracting relations from large plain-text collections. In: Proceedings of the 5th ACM International Conference on Digital Libraries, ACM, New York, NY, USA, June 2-7, pp: 85-94.
22. Evgeniy, G. and M. Shaul, 2004. Text categorization with many redundant features: Using aggressive feature selection to make SVMs competitive with C4.5. In: Proceedings of the 21st International Conference on Machine Learning, Banff, Canada, Morgan Kaufmann, Dec. 16-18, pp: 321-328.
23. Freitag and N. Kushmerick, 2000. Boosted wrapper induction. In: Proceedings of the 17th National Conference on Artificial Intelligence, AAAI Press The MIT Press, Sep. 11-18, pp: 577-583. ISBN: 0-262-51112-6.
24. Freitag and A. McCallum, 1999. Information extraction with HMMs and shrinkage. In Proceeding of the AAAI-99 Workshop on Machine Learning for Information Extraction, July 19-21. AAAI, Florida, pp: 31-36.

25. Guangzhi, Q., H. Salim and Y. Mazin, 2005. A new dependency and correlation analysis for features. IEEE Trans. Knowledge Data Eng., 17: 1199-1207. doi: 10.1109/TKDE.2005.136.
26. Hai Leong Chieu, 2003. Named entity recognition with a maximum entropy approach Named entity recognition with a maximum entropy approach. In: Proceeding of the 7th Conference on Natural Language Learning (CoNLL-2003), Association for Computational Linguistics Morristown, NJ, USA , May 31-June 1, pp : 160-163.
27. Hans Van Halteran, 2003. New feature sets for summarization by sentence extraction. IEEE Intelligent Syst., 18: 34-42. doi: http://doi.ieeecomputersociety.org/10.1109/MIS.2003.1217626.
28. Haralampos, K., T. Christos and T. Babis. An approach to text mining with information extraction. Centre Res. Inform. Manage., Manchester, UK.
29. John Abutridy, Chris Mellish and Stuart Aitken, May/June 2004. Combining Information extraction with genetic algorithm for text mining. IEEE Intel. Syst., 19: 22-30. doi: 10.1109/MIS.2004.4.
30. Lafferty, J., A. McCallum and F. Pereira, 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: Proceeding of the 18th ICML, June 28-July 1, Morgan Kaufmann, San Francisco, CA, pp: 282-289.
31. Raymond, J. Mooney and Razvan Bunescu, 2005. Mining knowledge from text using information extraction. ACM SIGKDD Explorati., 7: 3-10. doi: 10.3115/1067737.1067752.
32. Tom Betts, Maria Milosavljevic and Jon oberlander, The Utility of Information Extraction in the Classification of Books. Lecture Notes Comput. Sci., 4425: 295-306. doi: 10.1007/978-3-540-71496-5.