Natural Language Engineering **17** (2): 141–144. © Cambridge University Press 2011
doi:10.1017/S1351324911000015

Finite-state methods and models in natural language processing

ANSSI YLI-JYRÄ[1], ANDRÁS KORNAI[2] and JACQUES SAKAROVITCH[3]

[1]*Department of Modern Languages, PO Box 24, 00014 University of Helsinki, Finland*
email: anssi.yli-jyra@helsinki.fi

[2]*Computer and Automation Research Institute, Hungarian Academy of Sciences, Kende u 13-17,*
Budapest 1111, Hungary
and
Harvard University, Institute for Quantitative Social Science, 1737 Cambridge St,
Cambridge MA 02138, USA
email: andras@kornai.com

[3]*CNRS and Laboratoire Traitement et Communication de l'Information, Telecom ParisTech, 46,*
rue Barrault, 75634 Paris Cedex 13, France
email: sakarovitch@enst.fr

(*Received 20 December 2010*)

1 Introduction

For the past two decades, specialised events on finite-state methods have been successful in presenting interesting studies on natural language processing to the public through journals and collections. The FSMNLP workshops have become well-known among researchers and are now the main forum of the Association for Computational Linguistics' (ACL) Special Interest Group on Finite-State Methods (SIGFSM). The current issue on finite-state methods and models in natural language processing was planned in 2008 in this context as a response to a call for special issue proposals. In 2010, the issue received a total of sixteen submissions, some of which were extended and updated versions of workshop papers, and others which were completely new. The final selection, consisting of only seven papers that could fit into one issue, is not fully representative, but complements the prior special issues in a nice way. The selected papers showcase a few areas where finite-state methods have less than obvious and sometimes even groundbreaking relevance to natural language processing (NLP) applications.

These methods have grown around Kleene's classical result that relates languages defined by regular expressions to languages defined by finite automata (Kleene 1956). Practical finite-state methods apply and extend this correspondence. The most obvious extensions beyond the ordinary finite automata include finite transducers. Weighted automata provide another natural extension that potentially brings undecidability to the picture if weights are defined carelessly. However, typical definitions for weights are quite practical in applications as witnessed by the success of weighted transducers in automatic speech recognition. It should also be kept in mind that finite tree recognisers and transducers, and the corresponding tree grammars, have many finite-state characteristics as far as their tree languages are concerned.

Finite automata are particularly applicable because of their *closure properties*: various interesting methods of combining finite automata are guaranteed to yield finite automata. The closure properties are employed e.g. in information retrieval where the queries can be expanded and restricted with finite transducers (Koskenniemi 1996). Natural language question answering (QA) is a similar task (Hirschman and Gaizauskas 2001). **Mishra and Bangalore** describe a combination of a QA system and a spoken language recognition system and demonstrate nicely the use of weighted transducers in the interface between these.

One of the recurring ways in which finite-state methods prove themselves important is that they expand the possibilities of research by contributing fundamental algorithms to *digital libraries* and *specialised research infrastructures* that contain language resources and technology, such as the CLARIN initiative in Europe (www.clarin.eu). **Reffle**'s paper contains a finite-state algorithm that is useful when historical documents are prepared and processed for various kinds of linguistic inquiries. The work demonstrates that (1) finite-state algorithms help to improve the quality and the usability of the research material, (2) the methods used in research infrastructures need to be implemented with a lot of care and attention, and that (3) an on-demand combination of finite transducers is sometimes more efficient than an off-line compiled finite transducer.

String-based finite-state transducers are in fact quite commonly applied to *basic language processing* tasks such as text normalisation, morphological analysis and surface syntax. These applications were featured prominently in previous special issues of this journal (Kornai 1996; Karttunen, Koskenniemi and van Noord 2003), and over time we can discern some interesting trends. Most notably, *new open-source software* – to mention only OpenFST (www.openfst.org), SFST (www.ims.uni-stuttgart.de), Foma (foma.sourceforge.net) and HFST (hfst.sf.net) – now enable free basic language technology production for *under-resourced languages* using string-based methods. In this issue, Basque date expression normalisation is tackled by **Díaz de Ilarraza, Gojenola, Oronoz, and Alegria** whose experiments involve both proprietary and open-source finite-state tools.

At the bottom of the Chomsky hierarchy interesting subclasses of languages have been defined by considering subclasses of finite automata. The most remarkable one was characterised by Schützenberger (1965), who established that languages denoted by *star-free* regular expressions are the languages defined by *aperiodic* (or non-counting, or counter-free) finite automata. Aperiodicity is a property exhibited (with certain reservations) in an increasing number of linguistic domains and logics (Kornai 1985; Yli-Jyrä 2003, 2005; Droste and Gastin 2007; Hulden 2009; Rogers and Pullum to appear), not to mention Linear Temporal Logic (LTL), a popular tool in software verification. **Fernando**'s deep but exciting paper explores the conceptual issues arising when LTL and the associated model-theoretic semantics of time is adapted to natural language applications. This promises tools for *entailment relations* in the time domain that have applications to query expansion in QA systems.

Finite tree automata and tree grammars are a non-trivial extension of conventional finite-state methods. A good introduction to these models is in the TATA book (Comon et al. 2007). Regular tree grammars generalise over context-free grammars by separating the control of derivation from the labeling of the derived trees. Such separate control is employed e.g. in *constraints* that may be added to context-free grammars (Joshi and Levy 1982). Thanks to the added control mechanism of the extensions, a finite set of labelled trees can be regular (or recognisable) even if the set is not generated by any context-free grammar: the non-locality in the set is captured by state distinctions in a tree recogniser. **Högberg**'s paper gives an efficient algorithm for relaxing state distinctions in a way that is consistent with given positive and negative data. The algorithm could be applied e.g. to *supervised learning* from a treebank containing trees exhibiting wrong generalisations in addition to the usual trees that constitute the gold standard.

The *interface between syntax and semantics* is a key area where the linear structure of strings does not necessarily adhere to the argument structure and some sort of indication of the argument structure needs be assumed. Tree Adjoing Grammar (TAG), especially with feature structures, is reasonably good at capturing the syntax–semantics interface by the tree structures they assign to strings. **Gardent, Gottesman and Perez-Beltrachini** utilise the well-known relationship between the derivation trees of tree adjoining grammars and derived trees of regular tree grammars. The paper's experiments on natural language generation demonstrate nicely that regular tree grammars may be more efficient in practice, although they preserve the essential interface between syntax and semantics. The impact of such critical evaluation of methodologies can extend to various NLP tasks such as machine translation.

Finite-state methods have already had a significant impact on various approaches to statistical machine translation (SMT). As an interesting trend, tree-based finite-state methods have made their way to SMT. A well-articulated set of *desirable formal properties of tree transducers* have been a driving force in a lot of recent basic research resulting in a constantly evolving taxonomy of tree transducers (Knight 2007). In relation to this grid of desiderata, **Maletti** proposes an interesting tree transducer model that demonstrates the benefits of finite state control in capturing the non-local context, as opposed to the tree substitution grammar approach where the individual rules aim at capturing locally extended neighbourhoods in phrase-structure trees. Once again, the competitor is asymptotically less efficient than the finite-state approach presented here.

Acknowledgments

The position of the guest editors for this special issue was held jointly by András Kornai, Jacques Sakarovitch, and Anssi Yli-Jyrä. The guest editors would like to express their thanks to the authors, to the editor, Ruslan Mitkov, and the regular editorial board and staff of the journal for their help during the whole process. We thank the following for careful reviews: Julie Berndsen, Francisco Casacuberta, Jean-Marc Champarnaud, Jan Daciuk, Manfred Droste, Dafydd Gibbon, Colin de la Higuera, Lauri Karttunen, André Kempe, Kevin Knight, Hans-Ulrich Krieger, Marco Kuhlmann, Andreas Maletti, Ingmar Meinecke, Stoyan Mihov, Peter Mitankin, Mark-Jan Nederhof, Florent Nicart, Kemal Oflazer, Jakub Piskorski, Michael Riley, Strahil Ristov, Max Silberztein, Bruce Watson, Menno van Zaanen and those who wished to remain anonymous.

References

Comon, H., Dauchet, M., Gilleron, R., Löding, C., Jacquemard, F., D. Lugiez, Tison, S., and Tommasi, M. 2007. Tree automata techniques and applications. `http://www.grappa.univ-lille3.fr/tata`. Accessed on December 1, 2010.

Droste, M., and Gastin, P. 2007. Weighted automata and weighted logics. *Theoretical Computer Science* **380**: 69–86.

Hirschman, L., and Gaizauskas, R. 2001. Natural language question answering: the view from here. *Natural Language Engineering* **7**(4): 275–300.

Hulden, M. 2009. Regular expressions and predicate logic in finite-state language processing. In J. Piskorski, B. Watson, and A. Yli-Jyrä (eds.), *Finite-State Methods and Natural Language Processing. Post-Proceedings of the 7th International Workshop (FSMNLP) 2008*, pp. 82–97. Amsterdam, the Netherlands: IOS Press.

Joshi, A. K., and Levy, L. S. 1982. Phrase structure trees bear more fruit than you would have thought. *American Journal of Computational Linguistics* **8**(1): 1–11.

Karttunen, L., Koskenniemi, K., and van Noord, G. 2003. Special issue: finite-state methods in natural language processing. *Natural Language Engineering* **9**(1): 1–3.

Kleene, S. C. 1956. Representation of events in nerve nets and finite automata. In C. E. Shannon and J. McCarthy (eds.), *Automata Studies*, pp. 3–42. Princeton, NJ: Princeton University Press.

Knight, K. 2007. Capturing practical natural language transformations. *Machine Translation* **21**(2): 121–33.

Kornai, A. 1985. Natural language and the Chomsky hierarchy. In *Proceedings of the EACL 1985*, Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 1–7.

Kornai, A. 1996. Extended finite state models of language. *Natural Language Engineering* **2**(4): 287–90.

Koskenniemi, K. 1996. Finite-state morphology and information retrieval. *Natural Language Engineering* **2**(4): 331–6.

Rogers, J., and Pullum, G. K. (forthcoming) Aural pattern recognition experiments and the subregular hierarchy. To appear in Journal of Logic, Language and Information. A near-final draft accessed at http://www.lel.ed.ac.uk/∼gpullum/ on December 1, 2010. A shorter version presented to Mathematics of Language 10 Conference, UCLA, July 2007.

Schützenberger, M. P. 1965. On finite monoids having only trivial subgroups. *Information and Control* **8**(2): 190–4.

Yli-Jyrä, A. 2003. Describing syntax with star-free regular expressions. In *Proceedings of the EACL 2003*, Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 379–86.

Yli-Jyrä, A. 2005. Approximating dependency grammars through intersection of star-free regular languages. *International Journal of Foundations of Computer Science* **16**(3): 565–80.