

# *Finite-State Methods and Models in Natural Language Processing*

ANSSI YLI-JYRÄ

*University of Helsinki, Department of Modern Languages*

ANDRÁS KORNAI

*Hungarian Academy of Sciences Computer Science Research Institute and Harvard University*

JACQUES SAKAROVITCH

*CNRS and Telecom ParisTech, Laboratoire Traitement et Communication de l'Information*

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. A series of such events, the FSMNLP workshops, has become well-known among researchers and is now the main forum of SIGFSM, an ACL Special Interest Group. 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 16 submissions, some of which were extended and updated versions of workshop papers, and some which were completely new. The final selection, consisting of only seven articles that could fit into one issue, is not fully representative, but complements the prior special issues in a nice way. The selected articles showcase a few areas where finite-state methods have less than obvious and sometimes even groundbreaking relevance to natural language processing (NLP) applications.

Finite-state 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 most practical 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 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](http://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 et al., 2003), and over time we can discern some interesting trends. Most notably, *new open-source software* – to mention only OpenFST ([www.openfst.org](http://www.openfst.org)), SFST ([www.ims.uni-stuttgart.de](http://www.ims.uni-stuttgart.de)), Foma ([foma.sourceforge.net](http://foma.sourceforge.net)), and HFST ([hfst.sf.net](http://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.

Within the bottom class of Chomsky hierarchy characterised by finite automata, 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, 2010), 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 to conventional finite-state methods. A good introduction to these models is in the TATA book (Comon et al., 2007). To put it simply, regular tree grammars generalise over context-free grammars by separating the control of derivation from the labelling 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 labeled trees can be regular (or recognizable) 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 Adjoining Grammar (TAG), especially with feature structures, are 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 editor 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 to the whole process. We thank 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 Silberstein, Menno van Zaanen, Bruce Watson, and those who wished to remain anonymous.

### References

- Comon, H., M. Dauchet, R. Gilleron, C. Löding, F. Jacquemard, D. Lugiez, S. Tison, and M. Tommasi (2007). Tree automata techniques and applications. Available on: <http://www.grappa.univ-lille3.fr/tata>. October, 12th 2007.
- Droste, M. and P. Gastin (2007). Weighted automata and weighted logics. *Theoretical computer science* 380, 69–86.

- Hirschman, L. and R. Gaizauskas (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. IOS Press.
- Joshi, A. K. and L. S. Levy (1982). Phrase structure trees bear more fruit than you would have thought. *American Journal of Computational Linguistics* 8(1), 1–11.
- Karttunen, L., K. Koskenniemi, and G. van Noord (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), 121–133.
- Kornai, A. (1985). Natural language and the Chomsky hierarchy. In *Proceedings of the EACL 1985*, Geneva, Switzerland, pp. 1–7.
- Kornai, A. (1996). Extended finite state models of language. *Natural Language Engineering* 2(4), 287–290.
- Koskenniemi, K. (1996). Finite-state morphology and information retrieval. *Natural Language Engineering* 2(4), 331–336.
- Rogers, J. and G. K. Pullum (2010). Aural pattern recognition experiments and the subregular hierarchy. Manuscript. To appear in *Journal of Logic, Language and Information*.
- Schützenberger, M. P. (1965). On finite monoids having only trivial subgroups. *Information and Control* 8(2), 190–194.
- Yli-Jyrä, A. (2003). Describing syntax with star-free regular expressions. In *Proceedings of the EACL 2003*, Agro Hotel, Budapest, Hungary, pp. 379–386.
- Yli-Jyrä, A. (2005). Approximating dependency grammars through intersection of star-free regular languages. *International Journal of Foundations of Computer Science* 16(3), 565–580.