MODELLING NATURAL LANGUAGE WITH FINITE AUTOMATA

Karin Haenelt
Fraunhofer Gesellschaft e.V. (FhG)
Dolivostraße 15, 64293 Darmstadt

Introduction

Finite-state devices are used as a core technology in many fields of natural language processing. The applications include speech recognition and generation, spelling correction, fact extraction, information retrieval and approaches to translation. It has been shown that substantial aspects of natural languages can be processed with finite-state methods. Finite-state devices are very attractive for natural language processing, because they provide several advantages for language modelling as well as for mass data processing. The advantage for language modelling is that the framework allows for a uniform modelling of information which can be formulated with rules and of information which - due to the social and historical development of natural languages - is more word specific and must be lexicalized. In addition, weighted automata allow for accounting for the variability of data.
The advantages for mass data processing are manifold: finite-state automata are remarkably fast and they provide compressed representations of the data, which at the same time serve as search structures. Finite-state automata are mathematically well understood, and their algebraic foundation allows for a modular design and automatic compilation of system components. Due to the availability of high-level specifications they are easily maintainable and well-scalable. Modelling natural language with finite state automata has also provided much insight into structures and principles of natural languages and their complexity.

Three standard levels of natural language modelling and processing shall be looked at in more detail, namely lexical analysis (which for natural languages includes morphological analysis), syntactic analysis and semantic analysis. The automata used for these purposes are finite-state transducers which map an input language to an output language. The output language of one level serves as input language for the next level. The questions to be answered for the modelling are

- On which level and to what extent are natural languages regular languages?
- On which level and to what extent are regular languages adequate for natural language interpretation?
- How can mapping ambiguities between input languages and output languages be handled most efficiently?
- How can sources of non-determinism be reduced?
- How can finite-state technology be combined with other methods in order to process information which cannot be expressed with regular languages?

1. Lexical Analysis / Morphological Analysis: Words

Lexical analysis recognizes the basic units of natural language texts and maps them to normalized forms and linguistic features. Natural language words can be observed in texts, but different from programming language expressions they constitute a very large set, which - due to the productive character of word formation principles - is not closed. In addition they

occur in various forms, such as inflected forms (“*think*” -“ *thought*”), spelling variations and even in multiword groups (*“third countries”* - *“third-countries”*). With a few exceptions word formation of European languages can be modelled in terms of constrained concatenations of meaningful word components (morphs) (“*work-ing*”). The sets of meaningful word components (such as {*book, work*}, {*s, ed, ing*}) are represented in dictionaries, and their possible combinations are specified as concatenation conditions. The dictionaries are compiled into a finite-state transducer (lexical transducer) (Koskenniemi, 1983) (Beesley/Karttunen, 2003). The compilation includes the transformation of external specifications into automata, integration operations such as union or composition of transducers and optimisations. The resulting transducer is then used for lexical analysis. In order to keep the degree of non-determinisms of the lexical transducer low, some phenomena deserve special attention. These phenomena are mapping ambiguities between input and output languages, pattern overlapping with respect to the input language, non-disjoint morph classes in the concatenation rules, and long-distance dependencies. The methods used for modelling and optimisation range from pure finite-state approaches like determinisation (Mohri, 1996) to extensions of the finite-state framework: constraint propagation with bit-vectors (Barton/Berwick/Ristad, 1987) and context-free control structures.
For a better maintainability of the dictionaries high-level specifications of rules have been introduced which can be transformed into regular expressions and applied to the dictionaries by means of composition of rule and dictionary transducers (Karttunen, 1997).

2. Syntactic Analysis: Sequences of Words, Word Groups, Sentences

Syntactic analysis is concerned with the recognition of sequences of words and aims at attributing an interpreting structure to these sequences. The questions are

- Which sequences of natural language words are to what extent regular languages (sequences of words, word groups, sentences)?
- To what extent are regular languages adequate for interpretation?
- How can finite-state technology be combined with further approaches?

2.1 Characterisation of the situation

1. Which sequences of natural language words are to what extent regular languages?

It is generally agreed upon that not all phenomena of natural language sentences can be described with type-3 and type-2 grammars. This has been pointed out at first by Chomsky (1957). The following properties have been observed:
On the one hand

- there are concatenations which cannot be expressed with context-free languages ($x_1x_2 \ldots x_n \ldots y_1y_2 \ldots y_n$) (*Jan säit das mer (d'chind)^n (em Hans)^m es huus (lönd)^n (hälfe)^m aastriiche - John said that we the children-acc Hans-dat the house-acc let help paint*) (cf. Partee/ter Meulen/Wall, 1993)
- there are concatenations which cannot be expressed with regular languages
    - centre embedding ($S \rightarrow a\ S\ b$)
      (*the regulation which the commission (which the Council (which ...) had elected) had formulated*)
    - obligatorily paired correspondences or dependencies ($a^n a^R$)
      (*either ... or, if ... then*) which additionally can be nested inside each other

and on the other hand

- some local word ordering principles can be described in terms of regular languages
    - *“the good example” *”example good the”*

    - *"could have been shown" *"been could have shown"*
- some global word group ordering principles can be described in terms of regular languages
    - subject predicate object

There are, however, more structuring principles beyond concatenation which constrain word sequences in sentences and which are necessary for distinguishing well-formed from ill-formed sentences (case frames, world knowledge, pragmatics) (*we read a book, *we walk a book, colourless green ideas sleep furiously*). These principles may be even more important than concatenation.

2. To what extent are regular languages adequate for interpretation?

Natural language grammars have usually been written for recognition and parsing, i.e. their function is twofold:

- they are used for specifying allowable sequences of words, and
- they are used for assigning an interpretation structure to these sequences

As a consequence a weaker structure may be sufficient for recognition, but it may be regarded as inadequate for interpretation. Formally, grammars of type 2 can be transformed into grammars of type 3 (with the exception of centre embedding). But the transformations are only weekly equivalent, since they do not assign the same structure. In addition, they are not necessarily more efficient, since one context-free rule may have to be transformed in a whole complex of type-3 rules.

There are different approaches towards handling this situation. One line aims at developing grammar formalisms which are capable of handling all phenomena of a sentence. Another line aims at using the weakest possible and most efficient formalism for individual types of information.

2.2 Complete sentence parsing with mildly-context-sensitive formalisms

For full-sentence parsing more powerful formalisms have been developed. These formalisms (such as HPSG or tree adjoining grammars) allow for modelling further structuring principles (lexical principles, grammatical principles like subcategorisation, immediate dominance, head features) and they employ complex feature structures. In general, mildly-context-sensitive formalisms are regarded as adequate.

2.3 Partial sentence parsing with finite-state technology

Text parsing experience has shown that full-sentence grammars are neither robust nor efficient enough for mass data processing. This has lead to the development of shallow and partial parsing methods. These methods are based on finite-state technology.
Before viewing these approaches it may be helpful, to have a look at a sentence of a real text and to realise the requirements of sentence parsing:

1. The products referred to in paragraph 1 of point A may be held for sale or put on the market only in glass bottles which:
(a) are closed with:
- a mushroom-shaped stopper made of cork or other material permitted to come into contact with foodstuffs, held in place by a fastening, covered, if necessary, by a cap and sheathed in foil completely covering the stopper and all or part of the neck of the bottle,
- any other suitable closure in the case of bottles with a nominal content not exceeding 0,20

litres, and
(b) bear labelling conforming to the provisions of this Regulation.
(COUNCIL REGULATION (EC) No 1493/1999 of 17 May 1999 on the common organisation of the market in wine, Annex VIII G.)

This sentence follows several different structuring principles. The most obvious principles are

- concatenation of words in a specific order (a syntactic principle) and
- enumeration of statements (a textual principle).

Both principles are interwoven in this sentence. A closer look would reveal further structuring principles. But leaving these aside, it is important to notice that, in general, syntactic structures are hardly sufficient to describe the structures of whole sentences in real texts. This experience has already been summed up by Edward Sapir`s famous quote "All grammars leak" (1921), and it has been described by Manning and Schütze (1999:3) in the following way: "It is just not possible to provide an exact and complete characterization of all well-formed utterances that cleanly divides them from all other sequences of words, which are regarded as ill-formed utterances. … Nevertheless, it is certainly not the case that the rules are completely ill-founded. Syntactic rules for a language, such as that a basic English noun phrase consists of an optional determiner, some number of adjectives, and then a noun, do capture major patterns within the language. But somehow we need to make things looser, in accounting for the creativity of language use."

While Manning and Schütze motivate the use of statistical language models with their statement, we will view here the way partial parsing methods based on finite-state technology handle the interplay of syntactic and other principles in sentences and texts. Partial parsing methods clearly distinguish the individual composition principles, and syntactic parsing is confined to those parts that reliably can be recognized by syntactic criteria. These may be individual words, phrases or in some cases whole sentences. Fragments which do not follow the principles under analysis may be passed over. The technologies used include

- Pure finite state approach:
  reducing the recognition capacity and flattening the interpretation structure: It has been observed that structures which are more complex than regular sets - though being theoretically possible - do not occur very frequently in real texts. Obviously such structures are too complex even for human mental capacity. Corresponding approaches assume boundaries for centre embedding and model bounded centre embedding with type-3 grammars (Sproat, 2002).
  $S \rightarrow a\ S1\ b$, $S1 \rightarrow a\ S2\ b$, $S2 \rightarrow ab$, $S1 \rightarrow \varepsilon$, $S2 \rightarrow \varepsilon$.
- Context-free approximation without interpretation structure:
  A shift-reduce-recognizer is flattened into a finite-state acceptor with an input language which is a superset of L(G) defined by a context-free grammar G, and an output language which is confined to "accept" and "not accept" rather than attributing a structure (Pereira/Wright 1997). This method is employed for speech recognition where syntactic parsing serves to rank the hypotheses which are generated for the acoustic sequences.
- Context-free approximation with flattened interpretation structure: Finite-state cascades:
  Parsing with finite-state cascades is described by Abney (1995) as follows: "A finite-state cascade consists of a sequence of levels. Phrases at one level are built on phrases at the previous level, and there is no recursion: phrases never contain same-level or higher-level phrases. Two levels of special importance are the level of chunks and the level of simplex clauses …. Chunks are the non-recursive cores of "major" phrases, i.e., NP, VP, PP, AP, AdvP. Simplex clauses are clauses in which embedded clauses

have been turned into siblings — tail recursion has been replaced with iteration, so to speak." The interpretation structures are week equivalents of context-free structures, because they assume a fixed number of levels.

The processing advantages of partial and finite-state parsing are

- is robust, because it processes all phenomena which can be handled by the actual process and can pass over the phenomena outside its scope,
- it is very fast for two reasons
    - finite-state recognition is linear in time if deterministic automata can be used. It is still very fast if the degree of non-determinism can be kept low.
    - with non-recursive interpretation structures a source of heavy overgeneration of interpretation hypotheses of context-free grammars has been reduced. The heavy overgeneration is due to the fact that the phenomena involved (such as the attachment of prepositional phrases) are semantic in nature and are produced more or less brute-force as exhaustive lists of combinatorial possibilities by syntactic grammars, but cannot be decided on a syntactic basis. This has been a considerable factor in traditional parsing. Bod (1998) provides a sample sentence with 455 readings ("*List the sales of products produced in 1973 with the products produced in 1972*").

3. Semantic Analysis: Information Extraction

On the basis of lexical analysis and partial parsing even partial information processing is possible. Some approaches are

- *chunk linking and chunk attachment*
  Partial parsing can be extended by further interpretation steps which on the basis of semantic information such as verb case frames or corpus examples determine the linking of hitherto unconnected structures (such as the determination of subjects and objects of verbs or the attachment of some prepositional phrases)
- *finite state filtering*
  Grefenstette (1999) describes a layered finite-state parser which also groups adjacent syntactically related units and extracts non-adjacent n-ary grammatical relations. High level specifications of regular expressions are used for describing the patterns to be extracted.
- *head-modifier-pairs*
  In information retrieval robust parsing methods have been used for identifying head-modifier-pairs (such as [H: extract, M: information]). These pairs have been used for enriching the document index and to improve retrieval efficiency (Strzalkowski/Lin/Ge/Perez-Carballo, Jose (1999)).
- *fact extraction in fixed domains*
  Finite-state technology has also been applied to the extraction of information from highly standardized text types such as weather forecasts or stock market reports
- *message understanding*
  partial parsing and finite-state fact extraction has also been used in the competitions of the Message Understanding Conferences (MUC) where the task is filling in relational database templates from newswire texts. One of the approaches has been described by Hobbs/Appelt/Bear/Israel/Kameyama/Stickel/Tyson (1997). This approach uses a cascade of five transducers which include the recognition of names, fixed form expressions, basic noun and verb groups, patterns of events and the identification of event structures that describe the same event.

4. Summary

Natural languages are - due to their creative properties - potentially infinite (although there are neither words, nor sentences nor texts, which are really infinite). It is not possible to enumerate all phenomena; rather finite devices are needed for describing these languages with rules. For automatic processing the most adequate and least complex device is searched for. Finite-state devices can be characterised under these aspects as follows:

Linguistic adequacy

- Word formation of European languages can essentially be modelled in terms of regular languages.
- Sentence formation is theoretically of higher complexity than type-3 and type-2 grammars allow to model. But the more complex structures do not occur very often in real texts and are said to be psychologically too complex for practical use. Some of the frequent structures can be approximated with finite-state devices: The modelling of long distance dependencies can be approximated with feature propagation, centre embedding can be approximated be assuming boundaries, and tail recursion is more an interpretation phenomenon, than a structure which is needed for recognition.
- Sentences are assigned flat interpretation structures. These structures are linguistically adequate in the sense that they clearly identify the elements of sentence structures which are syntactic in nature. There are further structures and conditions which distinguish well-formed sentences from ill-formed sentences, but these principles are semantic and pragmatic in nature.
- Sentences are assigned only partial structures in many cases. These structures are linguistically adequate in the sense that sentences follow multiple structuring principles at the same time. These are syntactic, semantic and pragmatic principles which may complement one another, constrain each other and even overlap. The structures assigned by syntactic analysis are confined to those which are purely syntactic in nature and are not inadequately mixed with other structures.

Practical usefulness

- not all natural language phenomena can be described with regular devices
- many actually occurring phenomena can be described with regular devices
- not all practical applications require a complete and deep processing of natural language
- partial solutions allow for the development of many useful applications

Modelling devices

- finite state transducers with
    - input labels (natural language)
    - output labels (interpretation)
    - weights on transitions (for modelling the variability of the data and ranking the hypotheses)
    - additional output strings at final states (delayed output of ambiguities which as a result of determinisation have been pushed to the right)
- constraint propagation
- cascading of transducers for approximating context-free descriptions (functionally equivalent to composition of transducers, but without intermediate structure output; the individual transducers are considerably smaller than a composed transducer)

- high level specification for dictionaries and rules (regular expressions and replacement operators)
- lexicon compilers which transform high-level specifications into automata (compilation, union, composition, determinisation, minimisation)

Complexity

- Finite-state transducers can in many relevant cases be determinised and minimised
- Although the two-level model which maps an input language to an output language has in general been found to be computationally intractable (Barton/Berwick/Ristad, 1987), it can be observed, that in practice finite-state natural language systems do not involve complex search and are remarkably fast and well-suited for mass-data processing. It has been argued that the SAT-problem which has been used for demonstrating the intractability is unnatural. Natural language problems are bounded in size (input and output alphabets, word length of linguistic words, partiality of functions and relations) and combinatorial possibilities are locally restricted.

Mass data processing

The requirement of mass data processing has nearly necessarily lead to a re-discovery of finite-state technology, since only finite-sate devices are fast enough for this purpose. In the best case analysis is linear in time (for deterministic automata). Context-free grammars and mildly context-sensitive grammars have a cubic run-time complexity or even worse. Their real complexity is n^3 x |G|, where |G| is the size of the grammar, and $|G| >> n^3$ in systems with a relevant grammatical coverage.

5. References

Abney, Steven (1995). Partial Parsing via Finite-State Cascades. In: *Journal of Natural Language Engineering*, 2(4): 337-344.

Barton Jr., G. Edward; Berwick, Robert, C. und Eric Sven Ristad (1987). *Computational Complexity and Natural Language*. MIT Press.

Beesley, Kenneth R. and Lauri Karttunen (2003). *Finite-State Morphology.* Stanford, California: Center for the Study of Language and Information (Studies in Computational Linguistics).

Bod, Rens (1998). *Beyond Grammar. An Experienced-Based Theory of Language*. CSLI Lecture Notes, 88, Stanford, California: Center for the Study of Information and Language.

Chomsky, Noam (1957). Syntactic Structures. The Hague: Mouton.

Grefenstette, Gregory (1999). Light Parsing as Finite State Filtering. In: Kornai, András (ed.) Extended *Finite State Models of Language*. Cambridge: Cambridge University Press. Earlier version: Workshop on Extended finite state models of language, Budapest, Hungary, Aug 11--12, 1996. ECAI'96. http://citeseer.nj.nec.com/grefenstette96light.html

Hobbs, Jerry R.; Appelt, Douglas; Bear, John; Israel, David; Kameyama, Megumi; Stickel, Mark and Mabry Tyson (1997). FASTUS: A Cascaded Finite-State Transducer for Extracting Information from Natural Language Text. In: Roche, Emmanuel und Yves Schabes (Eds.) (1997). *Finite-State Language Processing.* Cambridge (Mass.) und London: MIT Press.

Karttunen, Lauri (1997). The Replace Operator. In: Roche, Emmanuel und Yves Schabes (Eds.) (1997). *Finite-State Language Processing.* Cambridge (Mass.) und London: MIT Press.

Kornai, András (ed.) (1999). *Extended Finite State Models of Language*. (Studies in Natural Language Processing). Cambridge: Cambridge University Press.

Koskenniemi, Kimmo (1983). *Two-level Morphology: a general computational Model for Word-form Recognition and Production.* Publication 11, University of Helsinki. Helsinki: Department of Genral Linguistics.

Kunze, Jürgen (2001). *Computerlinguistik. Voraussetzungen, Grundlagen, Werkzeuge.* Vorlesungsskript. Humboldt Universität zu Berlin. http://www2.rz.hu-berlin.de/compling/Lehrstuhl/Skripte/Computerlinguistik_1/index.html

Manning, Christopher D.; Schütze, Hinrich (1999). *Foundations of Statistical Natural Language Processing*. Cambridge, Mass., London: The MIT Press. (cf. http://www.sultry.arts.usyd.edu.au/fsnlp)

Mohri, Mehryar (1997). Finite State Transducers in Language and Speech Processing. In: *Computational Linguistics*, 23, 2, 1997, S. 269-311. http://citeseer.nj.nec.com/mohri97finitestate.html

Mohri, Mehryar (1996). On Some Applications of Finite-State Automata Theory to Natural Language Processing. In: *Natural Language Engineering*, 1996, vol. 2, no. 1, pp. 61-80. citeseer.nj.nec.com/mohri96some.html

Mohri, Mehryar und Michael Riley (2002). *Weighted Finite-State Transducers in Speech Recognition (Tutorial).* Teil 1: http://www.research.att.com/~mohri/postscript/icslp.ps, Teil 2: http://www.research.att.com/~mohri/postscript/icslp-tut2.ps

Partee, Barbara; ter Meulen, Alice and Robert E. Wall (1993). Mathematical Methods in Linguistics. Dordrecht: Kluwer Academic Publishers.

Roche, Emmanuel und Yves Schabes (Eds.) (1997). *Finite-State Language Processing.* Cambridge (Mass.) und London: MIT Press.

Sproat, Richard (2002). *The Linguistic Significance of Finite-State Techniques*. February 18, 2002. http://www.research.att.com/~rws

Strzalkowski, Tomek; Lin, Fang; Ge, Jin Wang; Perez-Carballo, Jose (1999). Evaluating Natural Language Processing Techniques in Information Retrieval. In: Strzalkowski, Tomek (Ed.): *Natural Language Information Retrieval*, Kluwer Academic Publishers, Holland : 113-145