T-(538|725)-MALV, Natural Language Processing Corpora and finite-state automata

Hrafn Loftsson\textsuperscript{1}  Hannes Högni Vilhjálmsson\textsuperscript{1}

\textsuperscript{1}School of Computer Science, Reykjavik University

August 2008
1. Corpora

2. Finite-state automata
1 Corpora

2 Finite-state automata
A corpus (í. málheild) is a collection of texts or speech stored in an electronic (machine-readable) format.

A corpus often contains material compiled using certain rules decided upon in advance.

Called a text collection (í. textasafn), rather than a corpus, if it contains randomly selected texts.

Huge corpora, tens (or hundreds) of millions of words, are available in many languages today.
## Types of corpora

### Genres
- Specific genres, e.g. law, science, novels, news text, etc.
- Wider variety of texts:
  - To survey comprehensively and accurately a language usage.
  - “Balancing a corpus”.
  - Costly task.
- Linguistic Data Consortium [http://www.ldc.upenn.edu/](http://www.ldc.upenn.edu/)

### Annotations
- Either, raw text without annotations, or
- Text with annotations (í. merkingar/skýringar).
Types of corpora

Genres

- Specific genres, e.g. law, science, novels, news text, etc.
- Wider variety of texts:
  - To survey comprehensively and accurately a language usage.
  - “Balancing a corpus”.
  - Costly task.
- Linguistic Data Consortium http://www.ldc.upenn.edu/

Annotations

- Either, raw text without annotations, or
- Text with annotations (í. merkingar/skýringar).
Corpora with annotations

What kind of annotations?

- Each word labeled with a linguistic tag (í. málfraðilegt mark)
- For example, part-of-speech (í. orðflokkur), constituent (í. setningarliður), semantic category (í. merkingarflokkur)
- Carried out manually and/or semi-automatically.

Treebank (í. trjábanki)

- A corpus, in which the syntactic structure of sentences is shown.
  - For example, a collection of parse trees.
- Penn Treebank (University of Pennsylvania) is probably the best known treebank.
Corpora with annotations

What kind of annotations?

- Each word labeled with a linguistic tag (í. málfræðilegt mark)
- For example, part-of-speech (í. orðflokkur), constituent (í. setningarliður), semantic category (í. merkingarflokkur)
- Carried out manually and/or semi-automatically.

Treebank (í. trjábanki)

- A corpus, in which the syntactic structure of sentences is shown.
  - For example, a collection of parse trees.
- Penn Treebank (University of Pennsylvania) is probably the best known treebank.
An example of a corpus

Penn Treebank

- http://www.cis.upenn.edu/~treebank/
- About 5 million words.
- POS tagged with a tagger.
  - http://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html
- Syntactically annotated with a parser.
An example from the Penn Treebank

POS tagged text

- The/DT grand/JJ jury/NN commented/VBD on/IN a/DT number/NN of/IN other/JJ topics/NNS ./.
- DT=Determiner(Article)=Ákvæðisorð (Greinir)
- JJ=Adjective=Lýsingarorð
- NN=Noun=Nafnorð
- VBD=Verb, past tense=Sögn í þátíð
- IN=Preposition or subordinating conjunction=Forsetning eða aukatenging
- NNS=Noun, plural=Nafnorð í fleirtölu
Pierre Vinken, 61 years old, will join the board as a nonexecutive director Nov. 29.
An example of a corpus

British National Corpus (BNC)

- http://www.natcorp.ox.ac.uk/
- 100 million words.
- A balanced corpus.
- Tagged with a tagger.
  - http://www.natcorp.ox.ac.uk/docs/bnc2postag_manual.htm
An example of a corpus

The Icelandic Frequency Dictionary (IFD) (í. Íslensk orðtíðnibók)

- \( \approx 590,000 \) tokens.
- A balanced corpus:
  - Icelandic fictions, translated fictions, biographies, educational material, children and teenager books.
- Tagged with a tagger (by Stefán Briem) and hand-corrected.
An example from the IFD

ég fp1\textsubscript{en} \hfill // word tag
stökk sfg1\textsubscript{eþ} \hfill // See explanation of tags
á aa \hfill // in a document under ‘‘Other material’’
eftir aþ \hfill // on the course web page
strætó nkeþ
og c
veifaði sfg1\textsubscript{eþ}
, ,
vagnstjórinn nkeng
sá sfg3\textsubscript{eþ}
mig fp1eo
og c
stoppaði sfg3\textsubscript{eþ}
An example of a corpus

A large Icelandic corpus

- Being compiled at The Árni Magnússon Institute of Icelandic studies (í. Stofnun Árna Magnússonar í íslenskum fræðum).
  - http://www.arnastofnun.is/page/arnastofnun_frontpage_en
- 900 text snippets, 25 million words.
- http://www.lexis.hi.is/malheild.htm
Utility of corpora

- The construction of word lists and dictionaries.
- Research in linguistics; corpus linguistics.
    ArticleURL&udi=B6VCH-3VTSB7V-1&user=5915045&_rdoc=1&_fmt=&_orig=search&_sort=d&view=c&_version=
    1&_urlVersion=0&_userid=5915045&md5=
    79f7bcbc4e2ed51ff8e73fdba9c06bd
- A precondition for the development of various LT tools, e.g.:
  - Taggers
  - Syntactic parsers
  - Machine translation systems (which often utilise parallel corpora).
Outline

1 Corpora

2 Finite-state automata
A device which accepts or rejects an input stream of tokens (i.e. strings).

Often called a “recognizer”.

Can also be used as a “generator”, i.e. a device which generates strings.

Very efficient in terms of speed and memory usage.

Very suitable for text searching.

Example: Fig. 2.1 page 28.
Finite-state automaton (FSA)

Mathematical definition

An FSA consists of five components \((Q, \Sigma, q_0, F, \delta)\):

1. \(Q\) is a finite set of states, \(q_0, q_1 \ldots q_n\).
2. \(\Sigma\) is a finite set of input symbols.
3. \(q_0\) is the start state, \(q_0 \in Q\).
4. \(F\) is the set of final states, \(F \subseteq Q\).
5. \(\delta\) is the transition function \(Q \times \Sigma \rightarrow Q\). \(\delta(q, i)\) returns the state to which the automaton moves when it is in state \(q\) and consumes the input symbol \(i\).

Example: \(Q = \{q_0, q_1, q_2\}, \Sigma = \{a, b, c\}, F = \{q_2\}, \delta = \{\delta(q_0, a) = q_1, \delta(q_1, b) = q_1, \delta(q_1, c) = q_2\}\)
Finite-state automaton (FSA)

**Mathematical definition**

An FSA consists of five components \((Q, \Sigma, q_0, F, \delta)\):

1. \(Q\) is a finite set of states, \(q_0, q_1 \ldots q_n\).
2. \(\Sigma\) is a finite set of input symbols.
3. \(q_0\) is the start state, \(q_0 \in Q\).
4. \(F\) is the set of final states, \(F \subseteq Q\).
5. \(\delta\) is the transition function \(Q \times \Sigma \rightarrow Q\). \(\delta(q, i)\) returns the state to which the automaton moves when it is in state \(q\) and consumes the input symbol \(i\).

Example: \(Q = \{q_0, q_1, q_2\}\), \(\Sigma = \{a, b, c\}\), \(F = \{q_2\}\), \(\delta = \{\delta(q_0, a) = q_1, \delta(q_1, b) = q_1, \delta(q_1, c) = q_2\}\)
Finite-state automata: Two types

**Deterministic Finite Automaton – DFA (í. Löggeng stöðuvél)**

Given a state and an input, there is one single possible destination state.

**Non-deterministic Finite Automaton – NFA (í. Brigðeng stöðuvél)**

- More than one path is possible from a state for an input.
- The path is not determined in advance.
- $\epsilon$ (the empty string) is an accepted input symbol.
- Example: Fig. 2.3 page 31.

An NFA can be converted to an equivalent DFA automatically.
Algorithm to simulate a DFA

- Input: a string $x$ ending with EOF. DFA, $D$, with start state $s_o$ and a set, $F$, of final states.
- Output: The answer “yes” if $D$ recognises $x$, otherwise “no”.

```plaintext
s = s_o
c = nextchar();
while (c <> EOF) {
    s = move(s, c);  // returns the state to which the automaton moves to from state s on input c
    c = nextchar();
}
if $s \in F$ then return “yes”
else return “no”;
```
Operations on Finite-State Automata

Main operations

- Union (í. Sammengi)
- Concatenation (í. Samtenging)
- Iteration; “Kleene Closure” (í. Endurtekning)

Union

- The union of two automata $A$ and $B$ accepts (or generates) all strings of $A$ and all strings of $B$.
- Denoted by $A \cup B$.
- Obtained by adding a new initial state with an $\epsilon$-transition to both $A$ and $B$ (See Fig. 2.7 page 34).
Operations on Finite-State Automata

Main operations

- Union (í. Sammengi)
- Concatenation (í. Samtenging)
- Iteration; “Kleene Closure” (í. Endurtekning)

Union

- The union of two automata $A$ and $B$ accepts (or generates) all strings of $A$ and all strings of $B$.
- Denoted by $A \cup B$.
- Obtained by adding a new initial state with an $\epsilon$-transition to both $A$ and $B$ (See Fig. 2.7 page 34).
Operations on Finite-State Automata

Concatenation

- The concatenation of two automata $A$ and $B$ accepts (or generates) all the strings that are concatenations of two strings, the first one being accepted by $A$ and the second one by $B$.
- Denoted $AB$.
- Obtained by connecting all the final states of $A$ to the initial state of $B$ using an $\epsilon$-transition (See Fig. 2.8 page 34).
### Operations on Finite-State Automata

#### Iteration

- “Closure” of an automaton $A$ accepts (or generates) the concatenations of any number of its strings and the empty string $\epsilon$.
- Denoted $A^*$. $A^* = \{\epsilon\} \cup A \cup AA \cup AAA \cup \ldots$
- Obtained by linking the final state of $A$ to its initial state using $\epsilon$-transition and adding a new initial state (See Fig. 2.9 page 34).
Operations on Finite-State Automata

Other common operations

- **Intersection** (í. Sniðmengi). The intersection of two automata $A \cap B$ accepts all the strings accepted both by $A$ and $B$.

- **Difference** (í. Mismunur). The difference of two automata $A - B$ accepts all the strings accepted by $A$ but not by $B$.

- **Complementation** (í. Uppbót?).
  - $\Sigma^*$ denotes the infinite set of all possible strings generated from the alphabet $\Sigma$.
  - The complementation of the automaton $A$ in $\Sigma^*$ accepts all the strings that are not accepted by $A$, i.e. $\hat{A} = \Sigma^* - A$. 
Transformations to optimize speed and memory requirements

- $\epsilon$-removal.
  - Transforms an initial automaton into an equivalent one without $\epsilon$-transitions.
- Determination.
  - Transforms an NFA to a DFA.
- Minimisation.
  - Constructs an equivalent automaton with as few states as possible.