# IceParser: An Incremental Finite-State Parser for Icelandic

Hrafn Loftsson[1] Eiríkur Rögnvaldsson[2]

[1]Department of Computer Science, Reykjavik University, Iceland
[2]Department of Icelandic, University of Iceland, Iceland

NoDaLiDa 2007

# Outline

# Outline

# Syntactic Analysis

## Full parsing

- Complete analysis (parse tree) is computed for each sentence.
- Disadvantages:
  - The set of solutions can grow exponentially.
  - The parser sometimes rejects a correct analysis of a sentence part.

## Shallow parsing

- Sentence parts or chunks are analysed without building a complete parse tree.
- The aim is "to recover syntactic information **efficiently and reliably** from unrestricted text, by sacrificing completeness and depth of analysis" (Abney, 1996).

# Syntactic Analysis

## Full parsing

- Complete analysis (parse tree) is computed for each sentence.
- Disadvantages:
  - The set of solutions can grow exponentially.
  - The parser sometimes rejects a correct analysis of a sentence part.

## Shallow parsing

- Sentence parts or chunks are analysed without building a complete parse tree.
- The aim is "to recover syntactic information **efficiently and reliably** from unrestricted text, by sacrificing completeness and depth of analysis" (Abney, 1996).

# Finite-state parsing

## Reductionist approach (Koskenniemi et al., 1992)

- Syntactic tags are associated with words.
- All possible readings of a sentence are reduced to one correct reading using elimination rules.

## Constructive approach

- Lexical description of a collection of syntactic patterns.
- Using a sequence of transducers, syntactic labels are inserted into the input strings, e.g.:
  - Brackets denoting constituent structure.
  - Names for grammatical functions.
- Xerox Finite-State Tool (XFST) (Karttunen et al., 1996)

# Finite-state parsing

## Reductionist approach (Koskenniemi et al., 1992)

- Syntactic tags are associated with words.
- All possible readings of a sentence are reduced to one correct reading using elimination rules.

## Constructive approach

- Lexical description of a collection of syntactic patterns.
- Using a sequence of transducers, syntactic labels are inserted into the input strings, e.g.:
  - Brackets denoting constituent structure.
  - Names for grammatical functions.
- Xerox Finite-State Tool (XFST) (Karttunen et al., 1996)

# Our motivation for developing a shallow parser

- No parser has been published for Icelandic.
- Shallow parsing is sufficient for many NLP applications, e.g.:
  - Information Extraction
  - Question answering
  - Some types of grammar checking
- Part of a BLARK (Basic Language Resource Kit)
- Efficiency is important $\Rightarrow$ Finite-state parser

# Outline

# The Icelandic language

## Heavily inflected

- Nouns: three genders, four cases, two numbers, sometimes suffixed definite article.
- Adjectives: four cases, three genders, two numbers, three degrees, “strong” and “weak” form.
- Verbs: three persons, two moods, two tenses, two voices.
- Word order is relatively free.

## The POS tagset

- Large, about 660 tags.
- Example: “hestarnir” (horses) $\Rightarrow$ *nkfng*; noun (n), masculine (k), plural (f), nominative (n), and suffixed definite article (g).

# The Icelandic language

## Heavily inflected

- Nouns: three genders, four cases, two numbers, sometimes suffixed definite article.
- Adjectives: four cases, three genders, two numbers, three degrees, "strong" and "weak" form.
- Verbs: three persons, two moods, two tenses, two voices.
- Word order is relatively free.

## The POS tagset

- Large, about 660 tags.
- Example: "hestarnir" (horses) $\Rightarrow$ *nkfng*; noun (n), masculine (k), plural (f), nominative (n), and suffixed definite article (g).

# Outline

# The annotation scheme

## Theory-neutral shallow annotation

- Constituent structure.
  - Standard labels: AdvP, AP, NP, PP, VP
  - Additionally: CP, SCP, InjP, MWE, APs, NPs
  - [NP ... NP], [VP ... VP]
  - [VPx ... VPx]; x $\epsilon$ {i, b, s, p, g}
- Functional tags.
  - Subjects and objects/complements: *SUBJ, *OBJ, *IOBJ, *OBJAP, *OBJNOM, *COMP
  - Other: *QUAL, *TIMEX
  - Relative position indicator, e.g.: *SUBJ>
    (the verb is positioned to the right of the subject)

# The annotation scheme

## A Grammar Definition Corpus

- Represents the major syntactic constructions in Icelandic.
- Examples:
  - {*SUBJ> [NP vagnstjórinn NP] *SUBJ>} [VP sá VP] {*OBJ< [NP mig NP] *OBJ<}
    (driver-the saw me)
  - {*SUBJ> [NP systir NP] {*QUAL [NP hennar NP] *QUAL} *SUBJ>} [VPb var VPb]
    (sister her was)
  - [VPb er VPb] {*SUBJ< [NP ég NP] *SUBJ<} {*COMP< [VPp fædd VPp] [CP og CP] [VPp uppalin VPp] *COMP<}
    (am I born and raised)

# Outline

# IceParser

## Design

- Designed to produce annotations according to our annotation scheme.
- A purely constructive finite-state parser.
- An incremental parser, which consists of two modules:
  - The *phrase structure module* (14 transducers).
  - The *syntactic functions module* (8 transducers).

## Implementation language

- Java and JFlex (a lexical analyser generator tool); the resulting Java code is a DFA.
- XFST is not used.

# IceParser

## Design

- Designed to produce annotations according to our annotation scheme.
- A purely constructive finite-state parser.
- An incremental parser, which consists of two modules:
  - The *phrase structure module* (14 transducers).
  - The *syntactic functions module* (8 transducers).

## Implementation language

- Java and JFlex (a lexical analyser generator tool); the resulting Java code is a DFA.
- XFST is not used.

# The transducers

- Include numerous syntactic patterns.
- And actions to add syntactic information into the text.
- Rely mainly on word class and subclass information from POS tags.
- Do not use the morphological information to full extent.
  - Only use the grammatical *case* feature.

# Outline

# The phrase structure module

- Input to first transducer is POS tagged text.
- Deepest constituents are analysed first; AdvP $\Rightarrow$ AP $\Rightarrow$ NP
- Adds brackets and labels to indicate constituent structure.
- Consider the patterns of the AP transducer:

```
Adj={WordSpaces}{AdjTag}
OpenAdvP="[AdvP"  CloseAdvP="AdvP]"
AdvPhrase={OpenAdvP}~{CloseAdvP}
AdjPhrase={AdvPhrase}?{Adj}
```

- [AdvP mjög aa AdvP] góður lkensf
  (very good)
  [AP [AdvP mjög aa AdvP] góður lkensf AP]

# The phrase structure module

- The NP transducer is the most complicated.
- Due to the free phrase-internal word order.
- The resulting DFA consists of about 50,000 states.
- [AP [AdvP mjög aa AdvP] góður lkensf AP] kennari
  (very good teacher)
  [NP [AP [AdvP mjög AdvP] góður AP] kennari NP]

# Outline

# The syntactic functions module

- Adds brackets and labels to indicate syntactic functions.
- Input to first transducer: Output of last transducer in the phrase structure module.
- Consider a part of the patterns of the COMP transducer:

```
Compl={APSeqNom}|{NPSeqNom} |
      {VPPastSeq}
SubjVerbBe={Subject}{WS}+{VPBe}{WS}+
SubjVerbCompl={SubjVerbBe}{Compl}
```

# The syntactic functions module

## An example

- {*SUBJ> [NP hann NP] *SUBJ>} [VPb er VPb] [NP [AP [AdvP mjög AdvP] góður AP] kennari NP]
- (he is (a) very good teacher)
- {*SUBJ> [NP hann NP] *SUBJ>} [VPb er VPb] [NP [AP {*COMP< [AdvP mjög AdvP] góður AP] kennari NP] *COMP<}

# Outline

# Evaluation

## Experimental setup

- A *gold standard* was constructed.
  - About 500 sentences randomly selected from the POS tagged *IFD* corpus.
  - Manually annotated with constituent structure and syntactic functions by two annotators using the annotation scheme.
- The *Evalb* bracket scoring program used for automatic evaluation.
- The parser evaluated using correct POS tags and tags generated by *IceTagger*.
  - POS tagging accuracy was 91.1% (unknown word ratio 7.8%).

# Results for the various phrase types

| Phrase type | F-measure using correct POS tags | F-measure using *IceTagger* | Freq. in test data |
|---|---|---|---|
| AdvP | 91.8% | 85.1% | 8.2% |
| AP | 95.1% | 86.3% | 8.1% |
| APs | 87.0% | 68.6% | 0.5% |
| NP | 96.8% | 93.0% | 37.6% |
| NPs | 80.4% | 74.3% | 1.5% |
| PP | 96.7% | 91.3% | 13.0% |
| VPx | 99.2% | 93.8% | 19.3% |
| CP | 100.0% | 99.6% | 5.7% |
| SCP | 99.6% | 97.6% | 3.4% |
| InjP | 100.0% | 96.3% | 0.2% |
| MWE | 96.9% | 92.6% | 2.5% |
| All | 96.7% | 91.9% | 100.0% |

# Constituents: A comparison

- First parser evaluation published for Icelandic.
- Comparison with Swedish, a related language:

| Parser | F-measure | | Tagger |
|---|---|---|---|
| | All phrases | NP | |
| IceParser | 96.7% | 96.8% | No |
| Kokkinakis & J.-Kokkinakis (1999) | 93.3% | 96.2% | Yes (98.7%) |
| IceParser | 91.9% | 93.0% | Yes (91.1%) |
| Knutsson et al. (2003)* | 88.7% | 91.4% | Yes |

* not finite-state

# Results for the various syntactic functions

| Function type | F-measure using correct POS tags | F-measure using *IceTagger* | Freq. in test data |
|---|---|---|---|
| SUBJ | 68.2% | 47.6% | 4.7% |
| SUBJ> | 92.7% | 89.4% | 30.3% |
| SUBJ< | 83.7% | 75.1% | 12.3% |
| OBJ | 0.0% | 0.0% | 0.2% |
| OBJ> | 43.5% | 20.0% | 0.8% |
| OBJ< | 90.2% | 78.2% | 19.7% |
| OBJAP> | 71.4% | 57.2% | 0.2% |
| OBJAP< | 75.0% | 46.2% | 0.4% |
| OBJNOM< | 30.8% | 16.7% | 0.6% |
| ... | | | |
| All | 84.3% | 75.3% | 100.0% |

# Results for the various syntactic functions (cont.)

| Function type | F-measure using correct POS tags | F-measure using *IceTagger* | Freq. in test data |
|---|---|---|---|
| ... | | | |
| IOBJ< | 73.3% | 51.9% | 0.9% |
| COMP | 56.9% | 40.0% | 2.8% |
| COMP> | 91.3% | 91.3% | 1.3% |
| COMP< | 75.1% | 70.0% | 12.7% |
| QUAL | 87.7% | 77.9% | 10.4% |
| TIMEX | 74.7% | 55.9% | 2.7% |
| All | 84.3% | 75.3% | 100.0% |

# Syntactic functions: A comparison

- Comparison with German, a related language:

| Parser | F-measure | | |
|---|---|---|---|
| | All functions | SUBJ | Tagger |
| IceParser | 84.3% | 90.5% | No |
| Müller (2004) | 82.5% | 90.8% | No |

# Efficiency

| Method | Word-tag pairs per sec. | Speed increase |
|---|---|---|
| Writing output to files | 6,700 | |
| Writing output to memory | 11,300 | 75% |

# Outline

# Error analysis

## AdvP

- Incorrect:
- [PP um [NP það NP] PP] [VP vissi VP] [NP stelpan NP] [AdvP ekki þá AdvP]
  (about that knew girl not then)
- [CP og CP] [VP tóku VP] [NP [AP [AdvP fram AdvP] eigin AP] dósir NP]
  (and took out own cans)
- Correct:
- [PP um [NP það NP] PP] [VP vissi VP] [NP stelpan NP] [AdvP ekki AdvP] [AdvP þá AdvP]
- [CP og CP] [VP tóku [AdvP fram AdvP] VP] [NP [AP eigin AP] dósir NP]

# Error analysis

## NP

- Incorrect:
- [NP árin NP] [AP gullnu AP]
  (years golden)
- Correct:
- [NP árin [AP gullnu AP] NP]

# Error analysis

## SUBJ

- Incorrect:
- [VPb er VPb] [AdvP ekki AdvP] [VPi að koma VPi]
  {*SUBJ [NP matur NP] *SUBJ} ?
  (is not to come food?)
- Correct:
- [VPb er VPb] [AdvP ekki AdvP] [VPi að koma VPi]
  {*SUBJ< [NP matur NP] *SUBJ<} ?

## OBJ

- {*OBJ< [NP [AP [AdvP fram AdvP] eigin AP] dósir NP]
  *OBJ<}

# Outline

# Summary

- IceParser is an incremental finite-state parser, based on a shallow annotation scheme.
  - A phrase structure module.
  - A syntactic functions module.
- F-measure: 96.7% for phrases, 84.3% for syntactic functions (assuming perfect tagging).
- Future work:
  - Improve individual components.
  - Build a version which uses the morphological info in POS tags to a greater extent.
- The parser can be tested at http://nlp.ru.is