Dynamic Programming of Partial Parses

David Cabrero Souto Jesús Vilares Ferro Manuel Vilares Ferro

Grupo CoLe

Universidad de Vigo & Universidade da Coruña
Introduction

The challenge of robust parsing:
Introduction

The challenge of robust parsing:

- incomplete grammars
Introduction

The challenge of robust parsing:
- incomplete grammars
- ill-formed input
Introduction

The challenge of robust parsing:

- incomplete grammars
- ill-formed input

...but there could be enough information.
Introduction

The challenge of robust parsing:

- incomplete grammars
- ill-formed input

...but there could be enough information.

Solutions:

- Automatic error recovery
- No parsing
- Partial parsing
Complete vs. Partial parsing

- Complete: only grammatical input
- Partial: relevant sub-parses

```
  complete
    a1 a2 a3 ... an
  partial
    a1 a2 a3 ... an
```
Approaches to partial parsing

Based on the idea of Chunking [Abney]

- First phase: parse minimal constituents (chunks)
- Second phase: join chunks

Modify conventional parsers
Approaches to partial parsing

Based on the idea of Chunking [Abney]
Approaches to partial parsing

- Based on the idea of Chunking [Abney]
  - n-phases (usually 2)
Approaches to partial parsing

Based on the idea of Chunking [Abney]
- n-phases (usually 2)
- First phase: parse minimal constituents (chunks)
Approaches to partial parsing

- Based on the idea of Chunking [Abney]
  - n-phases (usually 2)
  - First phase: parse minimal constituents (chunks)
  - Second phase: join chunks
Approaches to partial parsing

- Based on the idea of Chunking [Abney]
  - n-phases (usually 2)
  - First phase: parse minimal constituents (chunks)
  - Second phase: join chunks
- Modify conventional parsers
Modify conventional parsers

1. First of all: Extend grammars
Modify conventional parsers

1. First of all: Extend grammars
   
   CFG: $G = (N, \Sigma, P, S)$
Modify conventional parsers

1. First of all: Extend grammars

   CFG: $G = (N, \Sigma, P, S)$

   Noterminals
Modify conventional parsers

1. First of all: Extend grammars

   \[ \mathcal{G} = (N, \Sigma, P, S) \]

   Noterminals, Terminals
Modify conventional parsers

1. First of all: Extend grammars

- CFG: $\mathcal{G} = (N, \Sigma, P, S)$

  Noterminals, Terminals, Productions
Modify conventional parsers

1. First of all: Extend grammars

\[
\text{CFG: } G = (N, \Sigma, P, S)
\]

Noterminals, Terminals, Productions, Initial symbol
Modify conventional parsers

1. First of all: Extend grammars
   - CFG \( G = (N, \Sigma, P, S) \)
   - Noterminals, Terminals, Productions, Initial symbol
   - Change to: \( (N, \Sigma, P, S, S) \)
   - Set of Initial symbols
Modify conventional parsers

1. Extend grammars
   - CFG $\mathcal{G} = (N, \Sigma, P, S)$
     - Noterminals, Terminals, Productions, Initial symbol
   - Change to: $(N, \Sigma, P, S)$
     - Set of Initial symbols

2. Extend parser: Entry point [Jacobs]
   - Start at any point
   - Finish at any point
Descriptive Framework

Key idea: Parsing as deduction

Items
- Top-down

Deduction steps
- Top-down

Initial (axioms) and final (goals) items
- Top-down and
Descriptive Framework

Common descriptive framework [Sikkel, Shieber]
Descriptive Framework

- Common descriptive framework [Sikkel, Shieber]
- Key idea: Parsing as deduction
Descriptive Framework

- Common descriptive framework [Sikkel, Shieber]
- Key idea: Parsing as deduction
  - Items
    - Ex. Top-Down [$\bullet\beta, j$]
Descriptive Framework

- Common descriptive framework [Sikkel, Shieber]

- Key idea: Parsing as deduction
  - Items
    Ex. Top-Down \([\bullet \beta, j]\)
  - Deduction steps
    Ex. Top-Down \[\frac{\bullet B \beta, j}{\bullet \gamma \beta, j} \langle B \rightarrow \gamma \in R \rangle \]
    \[\frac{\bullet w_{j+1} \beta, j}{\bullet \beta, j + 1}\]
Descriptive Framework

- Common descriptive framework [Sikkel, Shieber]
- Key idea: Parsing as deduction
  - Items
    - Ex. Top-Down \([\bullet\beta, j]\]
  - Deduction steps
    - Ex. Top-Down \[
      \frac{\bullet B\beta, j}{\bullet \gamma \beta, j} \langle B \rightarrow \gamma \in R \rangle
    \]
    \[
    \frac{\bullet w_{j+1}\beta, j}{\bullet \beta, j + 1}
    \]
- Initial (axioms) and final (goals) items.
  - Ex. Top-Down \([\bullet S, 0]\) and \([\bullet, n]\)
Partial Top-Down

Axioms: \[ \bullet S, 0 \]
Goals: \[ \bullet, n \]
Partial Top-Down

Axioms: $[\bullet S, 0]$
Goals: $[\bullet, n]$

Change Initial Symbol ($S$) by set of Initial Symbols ($S$)
Partial Top-Down

Axioms: $[\bullet A, 0], A \in S$

Goals: $[\bullet, n]$

Change Initial Symbol ($S$) by set of Initial Symbols ($S$)
Partial Top-Down

Axioms: \([\bullet A, 0], A \in S\]

Goals: \([\bullet, n]\)

- Change Initial Symbol \((S)\) by set of Initial Symbols \((S)\)
- Start at any point
Partial Top-Down

Axioms: \([\bullet A, i], A \in \mathcal{S}, 0 \leq i \leq n\]

Goals: \([\bullet, n]\]

- Change Initial Symbol \(\mathcal{S}\) by set of Initial Symbols \(\mathcal{S}\)
- Start at any point
Partial Top-Down

Axioms: \([\bullet A, i], A \in \mathcal{S}, 0 \leq i \leq n\)

Goals: \([\bullet, n]\)

- Change Initial Symbol \((\mathcal{S})\) by set of Initial Symbols \((\mathcal{S})\)
- Start at any point
- Finish at any point
Partial Top-Down

Axioms: \([\bullet A, i], A \in \mathcal{S}, 0 \leq i \leq n\)

Goals: \([\bullet, j], 0 \leq j \leq n\)

- Change Initial Symbol \((\mathcal{S})\) by set of Initial Symbols \((\mathcal{S})\)
- Start at any point
- Finish at any point

Dynamic Programming of Partial Parses – p.7
Partial Top-Down

Axioms: $[\bullet A, i], A \in \mathcal{S}, 0 \leq i \leq n$

Goals: $[\bullet, j], 0 \leq j \leq n$

- Change Initial Symbol ($\mathcal{S}$) by set of Initial Symbols ($\mathcal{S}$)
- Start at any point
- Finish at any point
  1. Finish after starting
  2. Remember starting point
Partial Top-Down

Axioms: \[ [\bullet A, i, i], A \in S, 0 \leq i \leq n \]

Goals: \[ [\bullet, i, j], 0 \leq i \leq j \leq n \]

- Change Initial Symbol \((S)\) by set of Initial Symbols \((S)\)
- Start at any point
- Finish at any point
  1. Finish after starting
  2. Remember starting point
Partial Top-Down (2)

Next: modify deduction steps
Partial Top-Down (2)

- Next: modify deduction steps

Prediction

\[
\frac{[\bullet B\beta, j]}{[\bullet \gamma/\beta, j]} \quad \langle B \rightarrow \gamma \in R \rangle
\]

Scanning

\[
\frac{[\bullet w_{j+1}\beta, j]}{[\bullet \beta, j + 1]}
\]

- New in items: starting point
Partial Top-Down (2)

- Next: modify deduction steps

Prediction: 
\[
\frac{\bullet B\beta, j}{\bullet \gamma\beta, j} \quad \langle B \rightarrow \gamma \in R \rangle
\]

Scanning: 
\[
\frac{\bullet w_{j+1}\beta, j}{\bullet \beta, j + 1}
\]

- New in items: starting point
  - Add starting point
  - Pass-through starting point
Partial Top-Down (2)

- Next: modify deduction steps

Prediction
\[
\frac{\bullet B \beta, i, j}{\bullet \gamma \beta, i, j} \quad \langle B \rightarrow \gamma \in R \rangle
\]

Scanning
\[
\frac{\bullet w_{j+1} \beta, i, j}{\bullet \beta, i, j + 1}
\]

- New in items: starting point
  - Add starting point
  - Pass-through starting point
Experimental Results

Simple grammar: palindromes

![Graph showing the ratio (%) of input length against various parsing methods: Top Down, Bottom Up, Earley, and Dynamic LALR(1).]
Experimental Results (2)

- Results as expected
  - Top-Down: Combinatorial explosion
  - Bottom-Up: No more combinatorial explosion
  - LALR & Earley (mixed strategies): Good mix
Future work

- Real tests
- Proofs of soundness and correctness (TR ?)
- “Syntax sugar”++