Basic issues on Parsing

• Introduction
• Parsing issues
• Parsing CFG
• TN, RTN, ATN
• Charts
Basic issues on Parsing

- Parsing goals
  - Syntactic structure
  - Logic and basic semantic structure
- Syntax/semantics interaction
  - Only syntax
  - Only semantics
  - Performing in sequence
  - Performing in parallel
Basic issues on Parsing

- Parsing as searching in a search space
  - Characterizing the states
    - (if possible) enumerate them
  - Define the initial state (s)
  - Define (if possible) final states or the condition to reach one of them
Basic issues on Parsing

- Factors in parsing
  - Grammar expressivity
  - Coverage
  - Involved Knowledge Sources
  - Parsing strategy
  - Parsing direction
  - Production application order
  - Ambiguity management
  - (in)determinism
  - Parsing engineering
Basic issues on Parsing

“I was on the hill that has a telescope when I saw a man.”

“I saw a man who was on the hill that has a telescope on it.”

“I was on the hill when I used the telescope to see a man.”

“I saw a man who was on a hill and who had a telescope.”

“Using a telescope, I saw a man who was on a hill.”

.I . . .

I saw the man on the hill with the telescope

Taken from Loper
Basic issues on Parsing

- pattern matching
- TN => RTN => ATN
- WFST, Charts (M. Kay)
- Dynamic programming methods: CKY, Earley
- Phrase structure grammars: LSP (Sager), Diagram (Robinson)
- Deterministic parsers: LL, LR
- Parsifal (Marcus)
- Unification-based systems: DCG (Pereira, Warren), Patr II (Shieber)

A bit of history of Parsing

- Woods, 1970
- Kay, 1980
- Younger, 1967
- Earley, 1970
- Sager, 1981
- Robinson, 1982
- Aho et al, 1990
- Chapman, 1987
- Tomita, 1986, 1987
- Marcus, 1980
- Pereira, Warren, 1980
- Shieber, 1986
Basic issues on Parsing

- Parsers today
  - CFG (extended or not)
    - Tabular
    - Charts
    - LR
  - Unification-based
  - Statistical
  - Dependency parsing
  - Robust parsing (shallow, fragmental, chunkers, spotters)
Basic issues on Parsing

Parsing strategy

- Top Down
  - Guided by goals
  - Starts with a goal (or set of goals) to be built.
  - Tries to solve one of the pending goals
  - If more than one production can be applied:
    - Search problem
  - Pending goals can be reordered
  - Several search criteria (including heuristics) can be applied
  - The process ends when all the goals have been reached
Basic issues on Parsing

Parsing strategy

• Bottom up
  • Data driven
  • Starts from the sequence of words to be parsed (facts)
  • Proceeds bottom up
  • Several search criteria (including heuristics) can be applied
  • The process ends when the list of facts contains the initial symbol of the grammar.
Basic issues on Parsing

• Problems of TD strategy
  • Left recursivity
  • Many productions expanding the same non terminal
  • useless work
  • Search basically guided by the grammar
  • Repeated work
  • In general problems of backtracking algorithms
Basic issues on Parsing

- Problems of BU parsing
  - empty (optional) categories
  - Useless work (locally possible but globally impossible)
  - Inefficient when there is a high lexical ambiguity
  - Repeated work
ATN

• FSA -> Transition Network TN
  • States associated to the positions in the sentence
  • Arcs (transitions)
    • Labeled with POS
      • An arc can be traversed if the current word has the same POS as the arc.
  • Non determinism
    • More than one initial state
    • Current word with more than 1 POS
    • More than one arc for the same POS
ATN_2

\[ q_0 \xrightarrow{\text{det}} q_1 \xrightarrow{n} q_2 \xrightarrow{v} q_4 \xrightarrow{n} q_5 \xrightarrow{\text{det}} q_6 \]

\[ \text{the \ cat \ eats \ fish} \]

\[ \text{det \ n \ v \ n} \]
ATN \text{\textsubscript{3}}

TN limitations

- Only RG
- Only recognition
- Non-determinism $\Rightarrow$ backtracking
- No separation between grammar and parser
  - grammar $\Rightarrow$ syntactic model description
  - parser $\Rightarrow$ control
ATN 4

RTN

• Collection of TNs labeled with a name
  • Arcs
    • Labeled as in TN with POS
      • Terminal labels
    • Labeled with RTN identifiers
      • Non terminal labels
      • Final states in RTN produce coming back to the target state of the arc producing the call

• RTN are weakly equivalent to CFG
Sentence

NP

VP

1

2

3

NP

det

n

PP

adj

np

adj

n

adj

n

adj

n

adj

n

adj

n

adj

n
ATN 6

Graph representation of parsing rules:

1. **VP**
   - 1. **Vtrans**
   - 2. **NP**

2. **PP**
   - 1. **Prep**
   - 2. **NP**

Diagram:

```
  VP 1 Vtrans 2 NP
  PP 1 Prep 2 NP
```

NLP parsing general
RTN limitations

- Transitions depend only on the categories
  - CFG
- Only recognizing
- In fact fixed TD strategy
ATN

- Woods (1970)
- ATN = RTN with *operations* attached to arcs and use of *registers*.

- Operations
  - **Conditions**: Filter transitions between states
  - **Actions**: Building intermediate and output structures.
  - **Initializations**

- Allow expressing contextual constraints
Features
Number: Singular, Plural    Default: empty
Person: 1st, 2nd, 3rd         Default: 3rd
Rols: Subject

Taken from Winograd, 1983
Inicializations, Conditions and Actions

**NP-1:** \( f_{\text{Determiner}} \)
A: Set Number to the number of *

**NP-4:** \( g_{\text{Noun}} \)
C: Number is empty or number is the number of *
A: Set Number to the number of *
    Set Subject to *

**NP-5:** \( f_{\text{Pronoun}} \)
A: Set Number to the number of *
    Set Person to the Person of *
    Set Subject to *

**NP-6:** \( f_{\text{Proper}} \)
A: Set Number to the number of *
    Set Subject to *
ATN limitations

• Fixed TD strategy
• Redundancy in backtracking operations
• Problems of notational expressivity:
  • Very difficult to transport
Basic issues on Parsing

- Unified mechanism of parser description
  - Sikkel, 1997

- **Parser (schema):**
  - Given a sentence, an initial set of items is built
  - Given a grammar, a set of rules can be used for getting additional items

- **Parser (algorithm):**
  - Parsing schema
  - + data structures
  - + control structures
  - (+ communication structures)
• A *Chart* is a directed graph built dynamically along parsing
• Extension of WFST
• Nodes correspond to the start and end of the sentence and to the positions between words.
• Active arcs (goals or hypothesis) and inactive arcs (facts)
  • Notation active arcs: dotted rules
  • inactive arcs: category

0 1 2 3 4
• the • cat • eats • fish •
program chart
{ inicialize the chart with \( H \);
    inicialize the agenda with items which can be deduced without antecedents;
    while not empty (agenda)
    {extract current_item from agenda and put it on the chart;
     foreach item which could be deduced with one step including current_item
     {if item not in agenda and not in chart
         then add item to agenda
     }
    }
}
A concrete Chart algorithm should:
- define the structure of *agenda* and its scheduling criteria
- define order of performing deductive steps
- $D_{\text{scan}} + D_{\text{compl}} \rightarrow \text{Combination rule}$
- $D_{\text{pred}} \rightarrow \text{TD rule}$
- $\text{BU rule}$

BU strategy

TD strategy
Combination rule

When an active arc of the Chart reaches a node $j$ and from this node starts an inactive arc labeled with the category the active arc was waiting for, both arcs are combined for building a new arc (active or not) starting in the start node of the active arc and ending in the ending node of the inactive arc.
Charts

TD rule

When an active arc of the Chart reaches a node \( j \), for all the productions of the grammar expanding the category the active arc is waiting for a new active arc is built starting and ending in \( j \) corresponding to the dotted rule with dot in the initial position.

\[
\begin{align*}
B &\rightarrow \gamma \\
B &\rightarrow \eta \\
B &\rightarrow \varphi \\
&... \\
A &\rightarrow \alpha \bullet B \beta \\
B &\rightarrow \bullet \gamma \\
B &\rightarrow \bullet \eta \\
B &\rightarrow \bullet \varphi
\end{align*}
\]
BU rule

When an inactive arc of the Chart starts in a node $i$, for each production of the grammar owning as first constituent of the right side the category of the inactive arc a new active arc is built starting and ending in $i$ corresponding to the dotted rule with dot in the initial position.

```
B → Aγ
B → Aη
B → Aφ
...  
```
NLP parsing general
• Problems
  • The size of the Chart grows with the size of the grammar making the algorithm difficult to scale up.
  • A lot of useless active and inactive arcs are built.
  • In practice, lacking appropriate knowledge, a fixed BU strategy, eventually corrected with TD predictions, is used