UPPSALA
UNIVERSITET

Treebanks, evaluation, Earley Discussion

Syntactic analysis/parsing

2022

Sara Stymne
Department of Linguistics and Philology

Based on slides by Marco Kuhlmann

Treebank grammars

Reading rules off the trees

Given a treebank, we can construct a grammar by reading rules off the phrase structure trees.

Sample grammar rule	Span
S → NP-SBJ VP .	Pierre Vinken … Nov. 29.
NP-SBJ → NP , ADJP ,	Pierre Vinken, 61 years old,
VP → MD VP	will join the board …
NP → DT NN	the board

Properties of treebank grammars

- Treebank grammars are typically rather flat.
  Annotators tend to avoid deeply nested structures.
- Grammar transformations.
  In order to be useful in practice, treebank grammars need to be transformed in various ways.
- Treebank grammars are large.
  The vanilla PTB grammar has 29,846 rules.

Estimating rule probabilities

- The simplest way to obtain rule probabilities is relative frequency estimation.
- Step 1: Count the number of occurrences of each rule in the treebank.
- Step 2: Divide this number by the total number of rule occurrences for the same left-hand side.
- The grammar that you use in the assignment is produced in this way.

Parser evaluation

Evaluation measure

- **Precision:**
  Out of all brackets found by the parser,
  how many are also present in the gold standard?
- **Recall:**
  Out of all brackets in the gold standard,
  how many are also found by the parser?
- **F1-score:**
  harmonic mean between precision and recall:
  $2 \times \text{precision} \times \text{recall} / (\text{precision} + \text{recall})$

Evaluation and transformation

- It is good practice to always re-transform the grammar if it has been transformed, for instance into CNF
- In assignment 1 you will do your evaluation on the parse trees in CNF
  - It affects the scores, so they are not comparable to scores on the original treebank
  - This is not really good practice
  - But, it simplifies the assignment!

UPPSALA
UNIVERSITET

The Earley algorithm

CKY versus Earley

- The CKY algorithm has two disadvantages:
  - It can only handle restricted grammars.
  - It does not use top–down information.
- The Earley algorithm does not have these:
  - It can handle arbitrary grammars.
  - Is does use top–down information.
  - On the downside, it is more complicated.

The algorithm

- Start with the start symbol *S*.
- Take the leftmost nonterminal and predict all possible expansions.
- If the next symbol in the expansion is a word, match it against the input sentence (scan); otherwise, repeat.
- If there is nothing more to expand, the subtree is complete; in this case, continue with the next incomplete subtree.

Dotted rules

- A dotted rule is a partially processed rule.

  Example: S → NP • VP

- The dot can be placed in front of the first symbol, behind the last symbol, or between two symbols on the right-hand side of a rule.

- The general form of a dotted rule thus is $A \to \alpha \bullet \beta$, where $A \to \alpha\beta$ is the original, non-dotted rule.

Inference rules

Axiom

$$[0, 0, S \to \bullet\ \alpha]$$

$S \to \alpha$

Predict

$$\frac{[i, j, A \to \alpha \bullet B\ \beta]}{[j, j, B \to \bullet\ \gamma]}$$

$B \to \gamma$

Scan

$$\frac{[i, j, A \to \alpha \bullet a\ \beta]}{[i, j+1, A \to \alpha\ a \bullet \beta]}$$

$w_j = a$

Complete

$$\frac{[i, j, A \to \alpha \bullet B\ \beta] \quad [j, k, B \to \gamma\ \bullet]}{[i, k, A \to \alpha\ B \bullet \beta]}$$

Pseudo code 1

```
function EARLEY-PARSE(words, grammar) returns chart

  ENQUEUE((γ → • S, [0,0]), chart[0])
  for i ← from 0 to LENGTH(words) do
   for each state in chart[i] do
     if INCOMPLETE?(state) and
              NEXT-CAT(state) is not a part of speech then
        PREDICTOR(state)
     elseif INCOMPLETE?(state) and
              NEXT-CAT(state) is a part of speech then
        SCANNER(state)
     else
        COMPLETER(state)
   end
  end
  return(chart)
```

Pseudo code 2

procedure PREDICTOR($(A \rightarrow \alpha \bullet B\, \beta,\ [i,j])$)
    for each $(B \rightarrow \gamma)$ **in** GRAMMAR-RULES-FOR(B, *grammar*) **do**
        ENQUEUE($(B \rightarrow \bullet\, \gamma,\ [j,j])$, *chart*[$j$])
    end

procedure SCANNER($(A \rightarrow \alpha \bullet B\, \beta,\ [i,j])$)
    if B $\subset$ PARTS-OF-SPEECH(*word[j]*) **then**
        ENQUEUE($(B \rightarrow \textit{word}[j],\ [j, j+1])$, *chart[j+1]*)

procedure COMPLETER($(B \rightarrow \gamma\ \bullet,\ [j,k])$)
    for each $(A \rightarrow \alpha \bullet B\, \beta,\ [i,j])$ **in** *chart*[j] **do**
        ENQUEUE($(A \rightarrow \alpha\, B \bullet \beta,\ [i,k])$, *chart[k]*)
    end

Recogniser/parser

- When parsing is complete, is there a chart entry? [0, n, S → α •]
  - Recognizer
- If we want a parser, we have to add back pointers, and retrieve a tree
- Earley's algorithm can be used for PCFGs, but it is more complicated than for CKY

UPPSALA UNIVERSITET

Literature seminar 1, Feb 9

- *Recurrent neural network grammars*, Dyer, Kuncoro, Ballesteros, and Smith
- Detailed instructions on the course web page
  - Read the article carefully
  - Work through the given questions
  - Be prepared to discuss the article
- Make an effort to try to understand the paper!
- The seminar will help in understanding it!

UPPSALA UNIVERSITET

Literature seminars

- The seminar is obligatory and part of the examination
  - Use your full name when signing into Zoom (same link as for lectures)
  - Use your camera
    - For identification
    - To fascilitate discussion
- If you do not attend, have not prepared, or do not take part in the discussion: written report instead
- Groups and times available on the web page
- 45 minutes per group (5 students)

Coming sessions

- Wednesday:
  - 10.15: Advanced PCFG parsing Zoom lecture
  - 11-12: Supervision in Chomsky+Zoom
- Feb. 7: Supervision
- Feb 9: Literature seminar 1
- Graph-based dependency parsing: Feb 14
  - Fine to watch recordings afterwards

Coming assignments

- Assignment 1: February 11
- Assignment 2: March 4
  - Choose 2 articles, ask me to verify if you want to
  - Published articles, mainly on parsing, focus on algorithms
- Project:
  - Choose individual or pair project
    - Sign up in Studium
  - Decide on your topic
  - Proposal: February 25