Parsing:
Partial Parsing – Chunking

Partial Parsing

- For many applications you don’t really need a full-blown syntactic parse. You just need a good idea of where the base syntactic units are.
  - Often referred to as chunks.
- For example, if you’re interested in locating all the people, places and organizations in a text it might be useful to know where all the base noun phrases (NPs) are.
- A partial parse for just base NPs would be:

[$_{\text{NP}}$ Mr. Vinken] is [$_{\text{NP}}$ chairman] of [$_{\text{NP}}$ Elsevier N.V.],
[$_{\text{NP}}$ the Dutch publishing group].

Note that base NPs are the noun phrases that do not contain any other noun phrases.

Partial Parsing

- A full partial parse would have chunks for all types of phrases in the text, but with no hierarchical structure:

  $[_{NP}$ The morning flight$]$ $[_{PP}$ from$]$ $[_{NP}$ Denver$]$
  $[_{VP}$ has arrived$]$ $[_{PP}$ on$]$ $[_{NP}$ time$]$.

- For complete chunking, typical ordering:
  - Base syntactic phrases
  - Larger verb and noun groups
  - Sentential level rules, e.g. clauses

Rule-Based Partial Parsing

- With the lack of hierarchy between phrases and nesting within phrases (e.g. no NP can be inside another NP), parsing can be rule-based
  - Restrict the form of rules to exclude recursion (make the rules flat).
  - Group and order the rules so that the RHS of the rules can refer to non-terminals introduced in earlier rules but not later ones.
  - Write regular expressions to recognize the right-hand-side of rules, starting from the later ones.

NLTK Regular Expression Parsing

- Example chunk parser is NLTK’s regular expression parser
- Specify chunk phrases by giving regular expression patterns of POS tags
  - Example expression for noun phrases ending in common nouns:

```
NP: {<RB|DT|PP\$|PRP\$>?<JJ|JJR|JJS>*<VBN|VBG|NNP|CD>*<NN|NNS>+}
```

Matches noun phrases from the Penn Treebank like:

```
(NP asbestos/NN)
(NP a/DT fraction/NN)
(NP asbestos-related/JJ diseases/NNS)
(NP large/JJ burlap/NN sacks/NNS)
(NP 33/CD men/NNS)
(NP the/DT five/CD surviving/VBG workers/NNS)
(NP the/DT latest/JJS week/NN)
(NP six-month/JJ Treasury/NNP bills/NNS)
(NP The/DT average/JJ seven-day/JJ simple/JJ yield/NN)
(NP very/RB modest/JJ amounts/NNS)
```

NLTK Regular Expression Parsing

- Many types of noun phrases remain

```
(NP Terrence/NNP D./NNP Daniels/NNP)
(NP the/DT National/NNP Cancer/NNP Institute/NNP)
(NP New/JJ York-based/JJ Loews/NNP Corp./NNP)
(NP the/DT highest/JJS)
(NP it/PRP)
(NP who/WP)
(NP that/WDT)
(NP 1997/CD)
(NP 9.8/CD billion/CD)
```

- High scoring regex for all NP chunks

```
NP:
    {<DT>?<JJ|JJR|VBN|VBG>*<CD><JJ|JJR|VBN|VBG>*<NNS|NN>+}
    {<DT>?<JJS><NNS|NN>?}
    {<DT>?<PRP|NN|NNS><POS><NN|NNP|NNS>*}
    {<DT>?<NNP>+<POS><NN|NNP|NNS>*}
    {<DT|PRP\$>?<RB>?<JJ|JJR|VBN|VBG>*<NN|NNP|NNS>+}
    {<WP|WDT|PRP|EX>}
    {<DT><JJ>*<CD>}
    {<\$>?<CD>+}
```

Evaluation

- For evaluation, we need a metric that works at the level of the chunks.
- Precision:
  - The fraction of chunks the system returned that were right
    - "Right" means the boundaries and the label are correct given some labeled test set.
- Recall:
  - The fraction of the chunks that system got from those that it should have gotten.
- F measure: Combination of Precision and Recall