Last Class: Question-Answering Systems

Today: Probabilistic Parsing

1. Parsing with PCFGs
2. Problems
3. Probabilistic lexicalized CFGs

---

### CFG’s

A context free grammar consists of:

1. a set of non-terminal symbols $N$
2. a set of terminal symbols $\Sigma$ (disjoint from $N$)
3. a set of productions, $P$, each of the form $A \rightarrow \alpha$, where $A$ is a non-terminal and $\alpha$ is a string of symbols from the infinite set of strings ($\Sigma \cup N$)
4. a designated start symbol $S$

---

#### Probabilistic CFGs

Augments each rule in $P$ with a conditional probability:

$$A \rightarrow \beta \ [p]$$

where $p$ is the probability that the non-terminal $A$ will be expanded to the sequence $\beta$. Often referred to as

$$P(A \rightarrow \beta) \text{ or } P(A \rightarrow \beta|A).$$

---

#### Example

| $S$ $\rightarrow$ $NP$ $VP$ | [.80] |
| $S$ $\rightarrow$ $Aux$ $NP$ $VP$ | [.15] |
| $S$ $\rightarrow$ $VP$ | [.05] |
| $NP$ $\rightarrow$ $Det$ $Nom$ | [.20] |
| $NP$ $\rightarrow$ $Proper-Noun$ | [.35] |
| $NP$ $\rightarrow$ $Nom$ | [.05] |
| $NP$ $\rightarrow$ $Pronoun$ | [.40] |
| $Nom$ $\rightarrow$ $Noun$ | [.75] |
| $Nom$ $\rightarrow$ $Noun$ $Nom$ | [.20] |
| $Nom$ $\rightarrow$ $Proper-Noun$ $Nom$ | [.05] |
| $VP$ $\rightarrow$ $Verb$ | [.35] |
| $VP$ $\rightarrow$ $Verb$ $NP$ | [.40] |
| $VP$ $\rightarrow$ $Verb$ $NP$ $NP$ | [.05] |

| Det $\rightarrow$ $that$ | [.05] |
| Det $\rightarrow$ $the$ | [.80] |
| Det $\rightarrow$ $a$ | [.15] |
| Noun $\rightarrow$ $book$ | [.10] |
| Noun $\rightarrow$ $flights$ | [.50] |
| Noun $\rightarrow$ $meal$ | [.40] |
| Noun $\rightarrow$ $book$ | [.30] |
| Verb $\rightarrow$ $include$ | [.30] |
| Verb $\rightarrow$ $want$ | [.40] |
| Aux $\rightarrow$ $can$ | [.40] |
| Aux $\rightarrow$ $does$ | [.30] |
| Aux $\rightarrow$ $do$ | [.30] |
| Proper-Noun $\rightarrow$ $TWA$ | [.40] |
| Proper-Noun $\rightarrow$ $Denver$ | [.40] |
| Pronoun $\rightarrow$ $you$ | [.40] |
| Pronoun $\rightarrow$ $I$ | [.60] |
Why are PCFGs useful?

• Useful in disambiguation
  – Choose the most likely parse
  – Computing the probability of a parse
    If we make independence assumptions, \( P(T) = \prod_{n \in T} p(r(n)) \).

• Useful in language modeling tasks

Where do the probabilities come from?

1. from a treebank:
   \[ P(\alpha \rightarrow \beta | \alpha) = \frac{\text{Count}(\alpha \rightarrow \beta)}{\text{Count}(\alpha)} \]

2. use EM (forward-backward algorithm, inside-outside algorithm)

Parsing with PCFGs

Produce the most likely parse for a given sentence:

\[ \hat{T}(S) = \text{argmax}_{T \in \tau(S)} P(T) \]

where \( \tau(S) \) is the set of possible parse trees for \( S \).

• Augment the Earley algorithm to compute the probability of each of its parses.

When adding an entry \( E \) of category \( C \) to the chart using rule \( i \) with \( n \) subconstituents, \( E_1, \ldots, E_n \):

\[ P(E) = P(\text{rule } i \mid C) \cdot P(E_1) \cdot \ldots \cdot P(E_n) \]

• probabilistic CYK (Cocke-Younger-Kasami) algorithm
Problems with PCFGs

Do not model *structural dependencies*.

Often the choice of how a non-terminal expands depends on the location of the node in the parse tree.

E.g. Strong tendency in English for the syntactic subject of a spoken sentence to be a pronoun.

- 91% of declarative sentences in the Switchboard corpus are pronouns (vs. lexical).
- In contrast, 34% of direct objects in Switchboard are pronouns.

---

Problems with PCFGs

Do not adequately model *lexical dependencies*.

*Moscow sent more than 100,000 soldiers into Afghanistan...*

PP can attach to either the NP or the VP:

NP → NP PP or VP → V NP PP?

Attachment choice depends (in part) on the verb: *send* subcategorizes for a destination (e.g. expressed via a PP that begins with *into*).

---

Probabilistic lexicalized CFGs

- Each non-terminal is associated with its head.
- Each PCFG rule needs to be augmented to identify one rhs constituent to be the head daughter.
- Headword for a node in the parse tree is set to the headword of its head daughter.

---

Example

```
S(dumped)
  NP(workers)  VP(dumped)
    NNS(workers) VBD(dumped) NP(sacks) PP(into)
      NNS(sacks) P(into) NP(bin)
        DT(a) NN(bin)
```

workers dumped sacks into a bin
Probabilistic lexicalized CFGs

View a lexicalized (P)CFG as a simple (P)CFG with a lot more rules.

\[ \text{VP(dumped)} \rightarrow \text{VBD(dumped)} \text{ NP(sacks) PP(into)} \left[3 \times 10^{-10}\right] \]
\[ \text{VP(dumped)} \rightarrow \text{VBD(dumped)} \text{ NP(cats) PP(into)} \left[8 \times 10^{-10}\right] \]
\[ \text{VP(dumped)} \rightarrow \text{VBD(dumped)} \text{ NP(sacks) PP(above)} \left[1 \times 10^{-12}\right] \]

Problem?

Incorporating lexical dependency information

Incorporates lexical dependency information by:

1. relating the heads of phrases to the heads of their constituents;
2. including syntactic subcategorization information.

Syntactic subcategorization dependencies:

Probability of a rule \( r \) of syntactic category \( n \):
\[ p( r(n) \mid n, h(n) ) \]

Example: probability of expanding VP as \( \text{VP} \rightarrow \text{VBD} \text{ NP PP} \) will be
\[ p( \text{VP} \mid \text{VBD, dumped}) \]

Incorporating lexical dependency information

Condition the probability of a node \( n \) having a head \( h \) on two factors:

1. the syntactic category of the node \( n \)
2. the head of the node’s mother \( h(m(n)) \)
\[ p(h(n) = \text{word}_i \mid n, h(m(n))) \]

Computing the probability of a parse

Producing the most likely parse for a given sentence changes from:
\[ P(T) = \prod_{n \in T} p(r(n)) \]
to
\[ P(T) = \prod_{n \in T} p(r(n) \mid n, h(n)) \ast p(h(n) \mid n, h(m(n))) \]
Evaluation Measures and State of the Art

- labeled recall: \( \frac{\# \text{ correct constituents in candidate parse of } s}{\# \text{ correct constituents in treebank parse of } s} \)
- labeled precision: \( \frac{\# \text{ correct constituents in candidate parse of } s}{\text{total } \# \text{ of constituents in candidate parse of } s} \)
- crossing brackets: the number of crossed brackets

State of the art: 90% recall, 90% precision, 1% crossed bracketed constituents per sentence (WSJ treebank)