

Probabilistic Context Free Grammars

Von
Christian Voigtmann

Inhalt

- Definition Probabilistic CFG
- Wahrscheinlichkeit eines Satzes
- Bedingungen des PCFG
- Eigenschaften des PCFGs
- Zu klärende Fragen des PCFG
- „Inside probabilities“ / „Outside probabilities“
- Suche des besten „parse trees“
- Training vom PCFG
- Probleme des Inside-Outside Algorithmus

Definition Probabilistic CFG

Eine PCFG ist ein 5-Tupel $\langle w^k, N^i, N^1, R, P \rangle$

- Satz aus Terminalsymbolen, $\{w^k\}, k = 1, \dots, V$
- Satz aus Nicht-Terminalsymbolen, $\{N^i\}, i = 1, \dots, n$
- Startsymbol, N^1
- Satz aus Regeln, $N^i \rightarrow \zeta^j$
- Wahrscheinlichkeiten für die Regeln, $\forall i \sum_j P(N^i \rightarrow \zeta^j) = 1$

Wahrscheinlichkeit eines Satzes

$$P(w_{1m}) = \sum_t P(w_{1m}, t)$$

⇒ Die Variable „t“ ist hierbei ein geparster tree für einen Satz mit den Wörtern w_1 bis w_m

Bedingungen des PCFG

- **Place invariance:** Die Wahrscheinlichkeit eines „subtrees“ hängt nicht von seiner Position im Satz ab.
- **Context-free:** Die Wk eines „subtrees“ hängt nicht Wörtern außerhalb des „subtrees“ ab.
- **Ancestor-free:** Die Wk eines „subtrees“ hängt nicht von Knoten ab, die außerhalb des „subtrees“ ihre Herkunft haben.

Eigenschaften des PCFGs

- Mehrdeutigkeit: Je größer die Grammatik, desto mehr verschiedene Möglichkeiten entstehen einen Satz zu parsen.
- Einfache Modelle betrachten keine lexikalische „co-occurrence“.
- PCFGs können von ausschließlich positiven Beispielen lernen.
- Robustheit: Text kann grammatikalische Fehler enthalten; die Grammatik erkennt diese durch geringe Wk.

Eigenschaften des PCFGs

- Kleinere Bäume haben eine höhere Wk.
- Kombination der Stärken von PCFG mit Trigram-Modellen

Zu klärende Fragen des PCFG

1. Was ist die Wk eines Satzes in Abhängigkeit zur Grammatik G : $P(w_{1m} | G)$?
2. Welcher „parse“ für einen Satz, ist der mit der größten Wk?
3. Wie können die Wk der Regeln angepasst werden, damit die Wk eines Satzes maximiert wird? $\arg \max_G P(w_{1m} | G)$

„Inside probabilities“ / „Outside probabilities“

- Berechnung der Wk eines Strings:
 - Einfacher Ansatz; Aufsummierung der Wk aller „parse-trees“ eines Strings (ineffizient)!
 - Verwendung der „inside probabilities“ oder der „outside probabilities“ für die Berechnung der Wk eines Strings.

„Inside probabilities“ / „Outside probabilities“

- Die „inside probability“ ist all das, was von einem gegebenen Nicht-Terminalsymbol dominiert wird.

$$\beta_j(p, q) = P(w_{pq} \mid N^j_{pq}, G) = \dots = \dots$$

$$= \sum_{p, q, m} P(N^j \rightarrow N^r N^s) \beta_r(p, d) \beta_s(d+1, q)$$

- Die „outside probability“ ist all das, was außerhalb des Bereichs von p bis q ist.

$$a_j(p, q) = p(w_{1(p-1)}, N^j_{pq}, w_{(q+1)n}) = \dots = \dots$$

$$= \left[\sum_{f, g} \sum_{e=q+1}^m a_f(p, e) P(N^f \rightarrow N^j N^g) \beta_g(q+1, e) \right]$$

$$+ \left[\sum_{f, g} \sum_{e=1}^{p-1} a_f(e, q) P(N^f \rightarrow N^g N^j) \beta_g(e, p-1) \right]$$

Suche des besten „parse trees“

- Ähnlich wie der Viterbi-Algorithmus für HMM
- Unabhängigkeitsannahme gilt für PCFGs

$$\delta_i(p, p) = P(N^i \rightarrow w_p) \quad // \text{Initialisierung}$$

$$\delta_i(p, q) = \max_{\substack{1 \leq j, k \leq n \\ p \leq r < q}} P(N^i \rightarrow N^j N^k) \delta_j(p, r) \delta_k(r + 1, q) \quad // \text{Induktion}$$

$$\psi_i(p, q) = \arg \max_{(j, k, r)} P(N^i \rightarrow N^j N^k) \delta_j(p, r) \delta_k(r + 1, q) \quad // \text{Speicherung}$$

Training vom PCFG

- Versucht die optimalen W_k für die grammatikalischen Regeln zu finden!
- Dafür wird der Inside-Outside-Algorithmus verwendet.
- Parameter werden mit Hilfe unnotierter Sätze in der jeweiligen Sprache gelernt.

Probleme des Inside-Outside Algorithmus

- Verglichen mit linearen Modellen (HMM) ist das Modell langsamer.
- Lokale Maxima stellen öfter ein Problem da.
- Es werden für die Grammatik oft mehr Nicht-Terminalsymbole verwendet als nötig.