# K-best Iterative Viterbi Parsing

## K-best 迭代维特比句法分析

WeiYang

weiyang@godweiyang.com

www.godweiyang.com

East China Normal University
Department of Computer Science and Technology

2018.05.17

# Outline

# Motivations

- CKY or Viterbi inside algorithm is useful for PCFG parsing.
- However, parsing is slow when the grammar is large.
- Pruning techniques are often employed, such as beam search and coarse-to-fine search.
- However, pruning methods are approximate which can't always output the correct parsing trees.
- Iterative Viterbi Parsing (IVP) is used for pruning unnecessary edges which is much faster than CKY algorithm.

# Inside Algorithm

**Sum of inside potential:** $\alpha(A, i, j)$

**Initialization:**

If $A \rightarrow x_i \in R$, then $\alpha(A, i, i) = \varphi(A \rightarrow x_i, i, i, i)$, else 0.

**Bottom-up calculation:**

$$\alpha(A, i, j) = \sum_{A \rightarrow BC \in R} \sum_{k=i}^{j-1} \varphi(A \rightarrow BC, i, k, j) \cdot \alpha(B, i, k) \cdot \alpha(C, k+1, j)$$

# Outside Algorithm

**Sum of outside potential:** $\beta(A, i, j)$

**Initialization:**

$\beta(S, 1, n) = 1$, others 0.

**Top-down calculation:**

$$\beta(A, i, j) = \sum_{B \to AC \in R} \sum_{k=j+1}^{n} \varphi(B \to AC, i, j, k) \cdot \beta(B, i, k) \cdot \alpha(C, j+1, k)$$
$$+ \sum_{B \to CA \in R} \sum_{k=1}^{i-1} \varphi(B \to CA, k, i-1, j) \cdot \beta(B, k, j) \cdot \alpha(C, k, i-1)$$

# Outside Algorithm

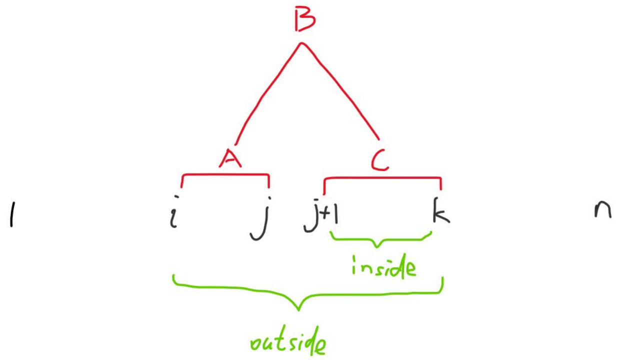


Figure: Outside Algorithm.

# Outside Algorithm

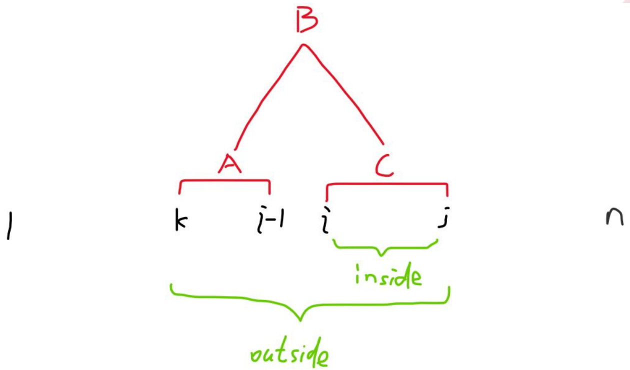


Figure: Outside Algorithm.

# Shrinkage Symbols

| Level | 0 | 1 | 2 |
|---|---|---|---|
| | Op_ | ADJ_ | JJ |
| | | | JJR |
| | | | JJS |
| | | ADV_ | RB |
| | | | RBR |
| | | | RBS |
| | | | WRB |
| | | NOUN_ | NN |
| | | | NNP |
| | | | NNPS |
| | | | NNS |
| | | VERB_ | MD |
| | | | VB |
| | | | VBD |
| | | | VBG |
| | | | VBN |
| | | | VBP |
| | | | VBZ |

Figure: The levels of non-terminal symbols.

# Chart Table

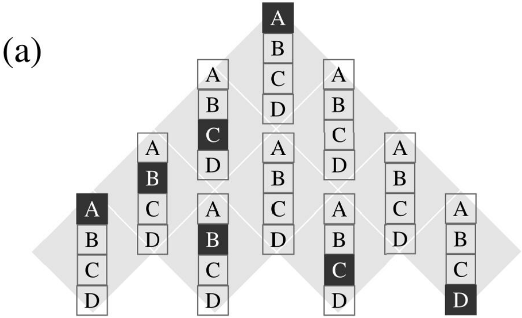


Figure: Original chart table consisting of non-terminal symbols only.

# Chart Table

(b)

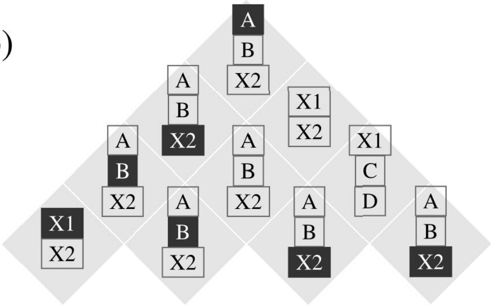


Figure: Coarse chart table consisting of both non-terminal symbols and shrinkage symbols.

# Definition

- Hierarchically cluster $N$ into $m+1$ sets $N_0 \dots N_m$ where $N = N_m$.
- Define a mapping $\pi_{i\to j} : N_i \mapsto \Im(N_j)$ where $\Im(N_j)$ is the power set of $\cdot$ .
- For $X_i \in N_i, X_j \in N_j, X_k \in N_k$, the rule parameter is defined as

$$\log q(X_i \to X_j X_k) = \max_{\substack{A\in\pi_{i\to m}(X_i)\\ B\in\pi_{j\to m}(X_j)\\ C\in\pi_{k\to m}(X_k)}} \log q(A \to BC)$$

- Each derivation in a coarse chart gives an upper bound on its corresponding derivation in the original chart.

# Lemma

If the best goal derivation $\hat{d}$ in the coarse chart does not include any shrinkage symbol, it is equivalent to the best goal derivation in the original chart.

**Proof:**

Let $\mathcal{Y}$ be the set of all goal derivations in the original chart, $\mathcal{Y}' \subset \mathcal{Y}$ be the subset of $\mathcal{Y}$ not appearing in the coarse chart, and $\mathcal{Y}''$ be the set of all goal derivations in the coarse chart. For each derivation $d \in \mathcal{Y}'$, there exists its unique corresponding derivation $d' \in \mathcal{Y}''$. Then, we have

$$\forall d \in \mathcal{Y}, \exists d' \in \mathcal{Y}'', s(d) \leq s(d') < s(\hat{d})$$

and this means that $\hat{d}$ is the best derivation in the original chart.

# Pseudo Code

**Algorithm 1** Iterative Viterbi Parsing

$lb \leftarrow \det(x, G)$ or $lb \leftarrow -\infty$
chart $\leftarrow$ init-chart$(x, G)$
**for all** $i \in [1 \ldots]$ **do**
    $\hat{d} \leftarrow$ Viterbi-inside(chart)
    **if** $\hat{d}$ consists of non-terminals only **then**
        return $\hat{d}$
    **if** $lb <$ best(chart) **then**
        $lb \leftarrow$ best(chart)
    expand-chart(chart, $\hat{d}$, $G$)
    Viterbi-outside(chart)
    prune-chart(chart, $lb$)

Figure: Iterative Viterbi Parsing.

# Pruning

- For an edge $e = (A, i, j)$, we denote by $\alpha\beta(e) = \alpha(e) + \beta(e)$ the score of the best goal derivation which passes through $e$
- If we obtain a lower bound $lb$ such that $lb \leq \max_{d \in \mathcal{Y}} s(d)$ where $\mathcal{Y}$ is the set of all goal derivations in the original chart, an edge $e$ with $\alpha\beta(e) < lb$ is no longer necessary to be processed.
- $\alpha\beta(e)$ can be efficiently computed by Viterbi inside-outside parsing its upper bound in a coarse chart table:

$$\alpha\beta(e) \leq \hat{\alpha}(e) + \hat{\beta}(e) = \hat{\alpha\beta}(e)$$

# K-best Extension

**Algorithm 2** K-best IVP

$lb \leftarrow$ beam$(x, G, k)$ or $lb \leftarrow -\infty$<br>
chart $\leftarrow$ init-chart$(x, G)$<br>
**for all** $i \in [1 \ldots]$ **do**<br>
    $\hat{d}_1 \leftarrow$ Viterbi-inside(chart)<br>
    **if** $\hat{d}_1$ consists of non-terminals only **then**<br>
        $[\hat{d}_2, \ldots, \hat{d}_k] \leftarrow$ Lazy K-best(chart)<br>
        **if** All of $[\hat{d}_2, \ldots, \hat{d}_k]$ consist of non-terminals only **then**<br>
            return $[\hat{d}_1, \hat{d}_2, \ldots, \hat{d}_k]$<br>
        **else**<br>
            $\hat{d}_1 =$ getShrinkageDeriv$([\hat{d}_2, \ldots, \hat{d}_k])$<br>
    **if** $lb <$ k-best(chart, $k$) **then**<br>
        $lb \leftarrow$ k-best(chart, $k$)<br>
    expand-chart(chart, $\hat{d}_1, G$)<br>
    Viterbi-outside(chart)<br>
    prune-chart(chart, $lb$)

Figure: K-best IVP.

# Experiments

| len. | CKY edges | CKY time | IVP edges | IVP pruned | IVP iters | IVP time |
|---|---|---|---|---|---|---|
| 20 | 10590 | 1.25 | 2864 | 2089 | 68 | 0.13 |
| 23 | 13938 | 1.76 | 2219 | 1462 | 41 | 0.06 |
| 22 | 12771 | 1.52 | 2204 | 1425 | 46 | 0.05 |
| 17 | 7701 | 0.72 | 1526 | 1119 | 32 | 0.03 |
| 28 | 20538 | 3.14 | 7306 | 5338 | 144 | 1.18 |
| 34 | 30141 | 5.44 | 6390 | 4634 | 98 | 0.49 |
| ⋮ | ⋮ | | | ⋮ | | |
| 21 | 12801 | 1.77 | 3502 | 2456 | 70 | 0.21 |

Figure: The number of the edges produced in 1-best parsing on testing set.

# Experiments

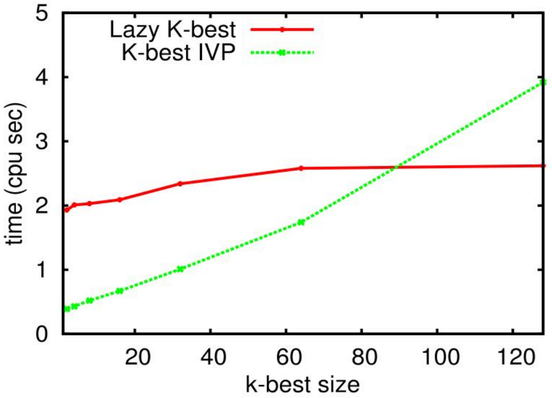


Figure: K-best Parsing time for various $k$.

# Experiments

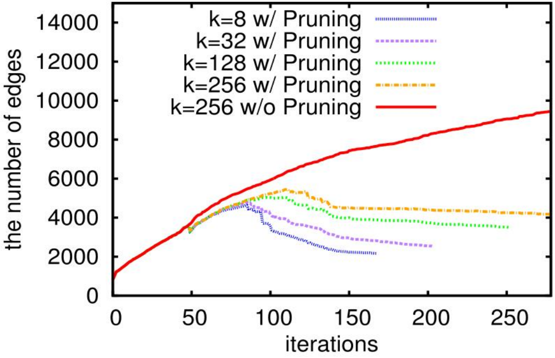


Figure: The plot of the number of edges in chart table at each K-best IVP parsing iteration.