A* Instead of Dynamic Programing

David McAllester
Toyota Technological Institute at Chicago (TTI-Chicago)

Pedro Felzenszwalb
University of Chicago

Dijkstra Lightest Parse

$$\begin{array}{l} \text{Weight}(X \Rightarrow y) = w \\ s_i = y \\ \rule{3em}{0.4pt}\ w \\ \text{Phrase}(X, i, i+1) = w \end{array}$$

$$\begin{array}{l} \text{Weight}(X \Rightarrow YZ) = w_1 \\ \text{Phrase}(Y, i, j) = w_2 \\ \text{Phrase}(Z, j, k) = w_3 \\ \rule{3em}{0.4pt}\ w_1 + w_2 + w_3 \\ \text{Phrase}(X, i, k) = w_1 + w_2 + w_3 \end{array}$$

Conclusions are kept on a priority queue at the priority given at the line of the inference rule.

A* CKY Parsing (Klein and Manning)

Let Γ be an abstraction function on nonterminals.

For example $\Gamma(NP_{\text{house}}) = NP$.

$$\frac{\text{Weight}(X \Rightarrow y) = w}{\text{Weight}(\Gamma(X) \Rightarrow y) = w}\ w \qquad \frac{\text{Weight}(X \Rightarrow YZ) = w}{\text{Weight}(\Gamma(X) \Rightarrow \Gamma(Y)\Gamma(Z)) = w}\ w$$

Computing a Heuristic Function

$$\frac{}{\text{Context}(\Gamma(S), 1, n) = 0}$$

$$\frac{\begin{array}{l}\text{Context}(\Gamma(X), i, k) = w_1 \\ \text{Weight}(\Gamma(X) \Rightarrow \Gamma(Y)\Gamma(Z)) = w_2 \\ \text{Phrase}(\Gamma(Y), i, j) = w_3\end{array}}{\text{Context}(\Gamma(Z), j, k) = w_1 + w_2 + w_3} \; w_1 + w_2 + w_3$$

$$\frac{\begin{array}{l}\text{Context}(\Gamma(X), i, k) = w_1 \\ \text{Weight}(\Gamma(X) \Rightarrow \Gamma(Y)\Gamma(Z)) = w_2 \\ \text{Phrase}(\Gamma(Z), j, k) = w_3\end{array}}{\text{Context}(\Gamma(Y), i, j) = w_1 + w_2 + w_3} \; w_1 + w_2 + w_3$$

A* Parsing

$$\text{Context}(\Gamma(X), i, k) = w_1$$
$$\text{Weight}(X \Rightarrow Y\,Z) = w_2$$
$$\text{Phrase}(Y, i, j) = w_3$$
$$\text{Phrase}(Z, j, k) = w_4$$
$$\overline{\qquad\qquad} \; w_1 + w_2 + w_3 + w_4$$
$$\text{Phrase}(X, i, k) = w_2 + w_3 + w_4$$

The General Case

Define an algorithm by rules of the following form.

$A_1 = w_1$

$\vdots$

$A_n = w_n$

$\overline{\qquad\qquad} \; w_1 + \ldots + w_n$

$C = w_1 + \ldots + w_n$

An Abstraction Hierarchy

Cconsider a sequence of abstraction spaces

$$\mathcal{H}_0 \xrightarrow{\Gamma} \mathcal{H}_1 \xrightarrow{\Gamma} \cdots \xrightarrow{\Gamma} \mathcal{H}_n$$

where $\mathcal{H}_n$ contains only a single assertion.

Levels can Function Independently

$$
\frac{\Gamma^i(A_1) = w_1 \quad \vdots \quad \Gamma^i(A_n) = w_n}{\Gamma^i(C) = w_1 + \ldots + w_n}\; w_1 + \ldots + w_n
$$

$$
\frac{\text{Context}(\Gamma^i(C)) = w_c \quad \Gamma^i(A_1) = w_1 \quad \vdots \quad \Gamma_i(A_n) = w_n}{\text{Context}(\Gamma_i(A_i)) = w_c + w_1 + \ldots + w_n - w_i}\; w_c + w_1 + \cdots + w_n
$$

$$
\frac{}{\text{Context}(\Gamma^i(Goal)) = 0}\; 0
$$

Contexts Give Admissible Heuristics

$$
\begin{array}{l}
\text{Context}(\Gamma^{i+1}(C)) = w_c \\
\Gamma_i(A_1) = w_1 \\
\vdots \\
\Gamma^i(A_n) = w_n \\
\hline
\Gamma^i(C) = w_1 + \ldots + w_n
\end{array}
\quad w_c + w_1 + \ldots + w_n
$$

Early Rules Part I

$$\text{Context}(\Gamma^i(goal)) = 0.$$

$$\text{Context}(\Gamma^i(C)) = w_c$$
$$\Gamma^{i+1}(A_1) = w_1$$
$$\Gamma^{i+1}(A_2) = w_2$$
$$\overline{} \; w_c + w_1 + w_2$$
$$\text{Context}(\Gamma^i(A_1)) = w_c + w_2$$

Early Rules Part II

$$\begin{array}{l} \text{Context}(\Gamma^i(C)) = w_c \\ \Gamma^i(A_1) = w_1 \\ \Gamma^{i+1}(A_2) = w_2 \\ \overline{\hphantom{xxxxxx}}\ w_c + w_1 + w_2 \\ \text{Context}(\Gamma^i(A_2)) = w_c + w_1 \end{array}$$

$$\begin{array}{l} \text{Context}(\Gamma^i(C)) = w_1 \\ \Gamma^i(A_1) = w_2 \\ \Gamma^i(A_2) = w_3 \\ \overline{\hphantom{xxxxxx}}\ w_c + w_1 + w_2 \\ \Gamma^i(C) = w_1 + w_2 \end{array}$$

The Pipeline

acoustic signal

$\longrightarrow$ word sequence

$\longrightarrow$ parse tree

$\longrightarrow$ shallow semantics

$\longrightarrow$ coreference

$\longrightarrow$ semantic entailment