Conditions on Consistency of Probabilistic Tree Adjoining Grammars

COLING/ACL 1998

Anoop Sarkar
Dept. of Computer and Information Science
University of Pennsylvania
anoop@linc.cis.upenn.edu
Consistency of Probabilistic Grammars

- Pr assigns a probability to each string \( v \) in the language.

- If \( v \) is not in the language then \( \Pr(v) = 0 \).

- Consistency is the property that sum of probabilities assigned to all the strings in the language sum to 1.

\[
\sum_{v \in L(G)} \Pr(v) = 1
\]
Probabilistic TAGs
$t_3$ $a_3$ $B^*$ $A_2$

$B_1$ $A$

$A_3$ $a_1$

$a_2$

$\phi(A_1 \leftrightarrow t_2) \times \phi(B_1 \leftrightarrow t_3)$
\[ \phi(A_1 \mapsto t_2) \times \phi(B_1 \mapsto t_3) \]
\[ \times \phi(A_2 \mapsto nil) \times \phi(A_3 \mapsto nil) \]
\[ \times \phi(B_2 \mapsto nil) \]
\[
\phi(S_1 \mapsto t_2) = 1.0
\]

\[
\begin{align*}
\phi(S_2 \mapsto t_2) &= 0.99 \\
\phi(S_2 \mapsto \text{nil}) &= 0.01 \\
\phi(S_3 \mapsto t_2) &= 0.98 \\
\phi(S_3 \mapsto \text{nil}) &= 0.02
\end{align*}
\]
TAG Derivations and Branching Processes

• There is an initial set of objects in the 0-th generation which produces with some probability a first generation.

• The first generation in turn with some probability generates a second, and so on.

• We will denote by vectors $Z_0, Z_1, Z_2, \ldots$ the 0-th, first, second, $\ldots$ generations.
TAG Derivations and Branching Processes

- The size of the $n$-th generation does not influence the probability with which any of the objects in the $(n + 1)$-th generation is produced.

- $Z_0, Z_1, Z_2, \ldots$ form a Markov chain.

- The number of objects born to a parent object does not depend on how many other objects are present at the same level.

- We associate a generating function for each level $Z_i$. 
Adjunction Generating Function

\[ g_1(s_1, \ldots, s_5) = \phi(A_1 \mapsto t_2) \cdot s_2 \cdot s_3 \cdot s_4 + \phi(A_1 \mapsto nil) \]
Level generating functions

\[
G_0(s_1, \ldots, s_k) = s_1
\]
\[
G_1(s_1, \ldots, s_k) = g_1(s_1, \ldots, s_k)
\]
\[
G_n(s_1, \ldots, s_k) = G_{n-1}[g_1(s_1, \ldots, s_k), \ldots, g_k(s_1, \ldots, s_k)]
\]

- we can express \(G_i(s_1, \ldots, s_k)\) as a sum \(D_i(s_1, \ldots, s_k) + C_i\)

- A probabilistic TAG will be consistent if these recursive equations terminate, i.e. iff

\[
\lim_{i \to \infty} D_i(s_1, \ldots, s_k) \to 0
\]
\[
\begin{align*}
\phi(A_1 \mapsto t_2) &= 0.8 \\
\phi(A_1 \mapsto \text{nil}) &= 0.2
\end{align*}
\]
\[
P = \begin{bmatrix}
A_1 & 0 & 0.8 & 0 \\
A_2 & 0 & 0.2 & 0 \\
B_1 & 0 & 0 & 0.2 \\
A_3 & 0 & 0.4 & 0 \\
B_2 & 0 & 0 & 0.1
\end{bmatrix}
\]

\[
N = \begin{bmatrix}
A_1 & A_2 & B_1 & A_3 & B_2 \\
t_1 & 1.0 & 0 & 0 & 0 & 0 \\
t_2 & 0 & 1.0 & 1.0 & 1.0 & 0 \\
t_3 & 0 & 0 & 0 & 0 & 1.0
\end{bmatrix}
\]
\[ \mathcal{M} = P \cdot N = \begin{bmatrix}
A_1 & A_2 & B_1 & A_3 & B_2 \\
A_1 & 0 & 0.8 & 0.8 & 0.8 & 0 \\
A_2 & 0 & 0.2 & 0.2 & 0.2 & 0 \\
B_1 & 0 & 0 & 0 & 0 & 0.2 \\
A_3 & 0 & 0.4 & 0.4 & 0.4 & 0 \\
B_2 & 0 & 0 & 0 & 0 & 0.1
\end{bmatrix} \]

- By representing the TAG derivations as a (Markovian) branching process we obtain a convergence result for \( \mathcal{M} \).

- This allows us to test for consistency of the probabilistic TAG by computing \( \text{eig}(\mathcal{M}) \).

- In our example, the eigenvalues are 0, 0, 0, 6, 0, and 0.1. Since all are less than 1 the grammar is consistent.
Summary

- Derived conditions under which given probabilistic TAG can be shown to be consistent.

- Gave a simple algorithm for checking consistency.

- Gave formal justification for correctness of the algorithm.

- Useful for checking deficiency in a probabilistic TAG.