
Loosely Tree-Based Alignment for Machine Translation

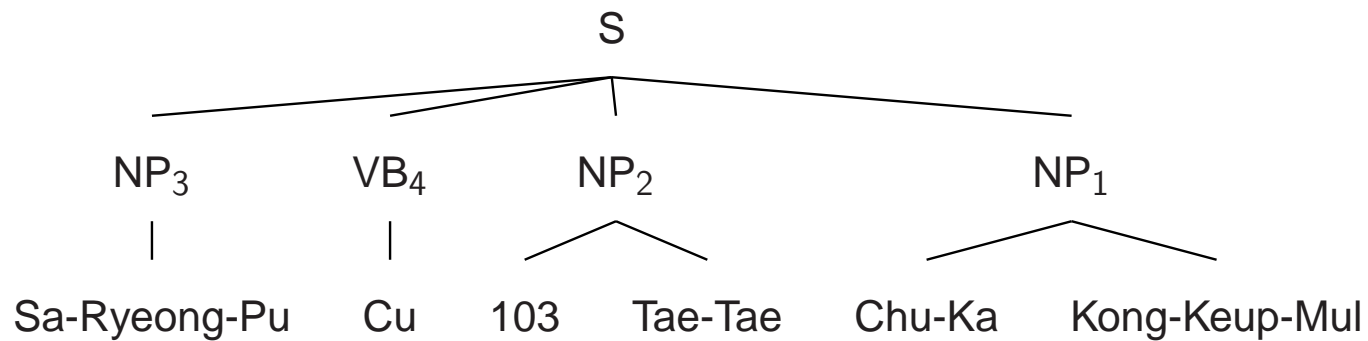
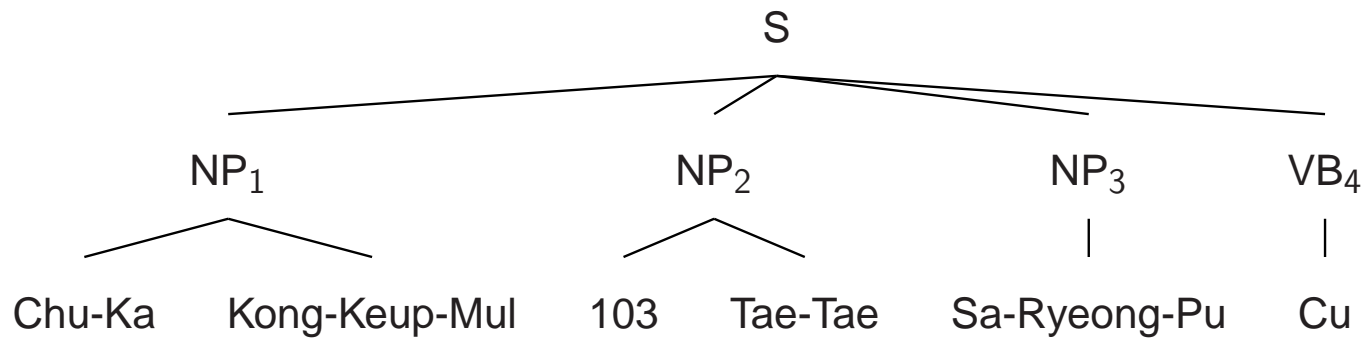
Daniel Gildea
University of Pennsylvania

Structured/Syntax-Based Alignment Models

- Stochastic Inversion Transduction Grammars (Wu 1997)
- Collections of Finite State Head Transducers (Alshawi, Bangalore, Douglas 2000)
- Tree-to-String transformation (Yamada and Knight 2001, 2002)
- This talk: Non-isomorphic transformations of human annotated trees
 - compare use of parse tree from one vs. two languages

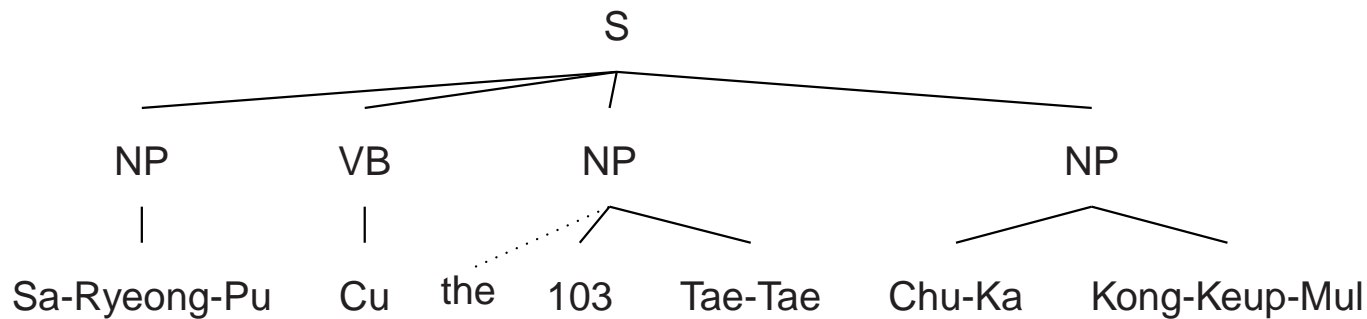
Tree-Based Alignment

Yamada & Knight 2001

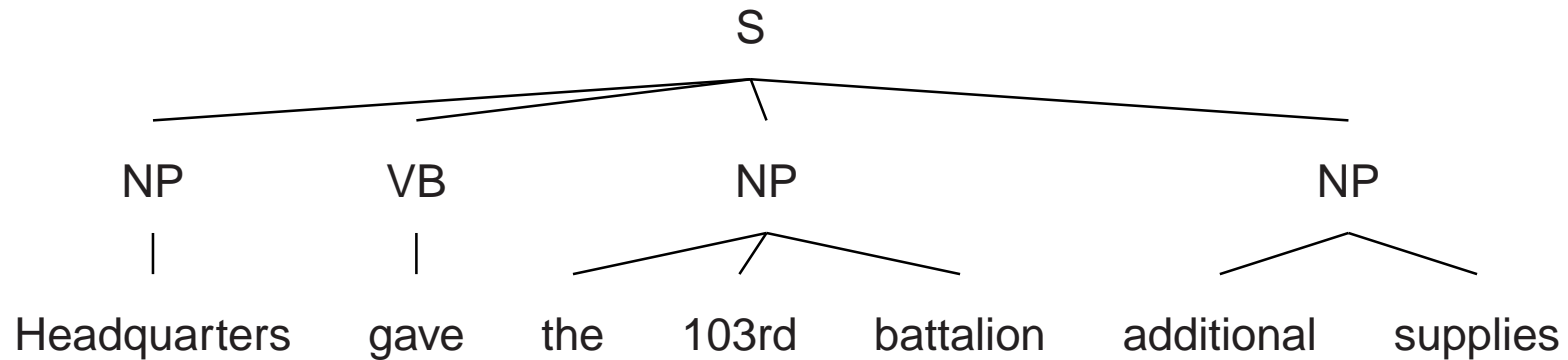


re-order step: $P_r(3, 4, 2, 1 \mid S \Rightarrow NP \ NP \ NP \ VB)$

Tree-Based Alignment 2

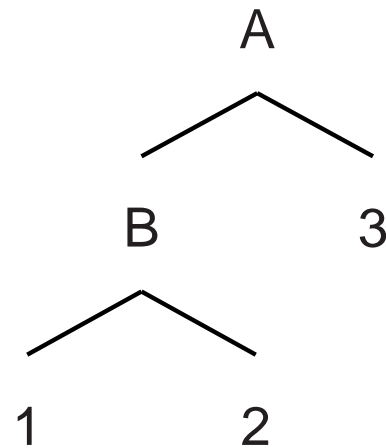


insertion step: $P_{ins}(\text{the})P(\text{ins}|NP)$



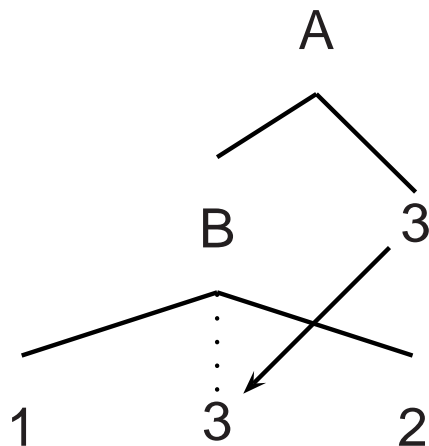
translation step: $P_t(\text{give}|Cu)$

Trees Constrain Possible Alignments



Of the six possible re-orderings of the three terminals, the two are not allowed: 1,3,2 and 2,3,1

Allow Subtrees to be “Cloned”



Constituents of sentence can move to arbitrary locations, at a cost in probability.

Parameterization

Assumption that individual clone operations are independent means no increase in computational complexity.

Probability of inserting a copy of node ε_i as a child of ε_j :

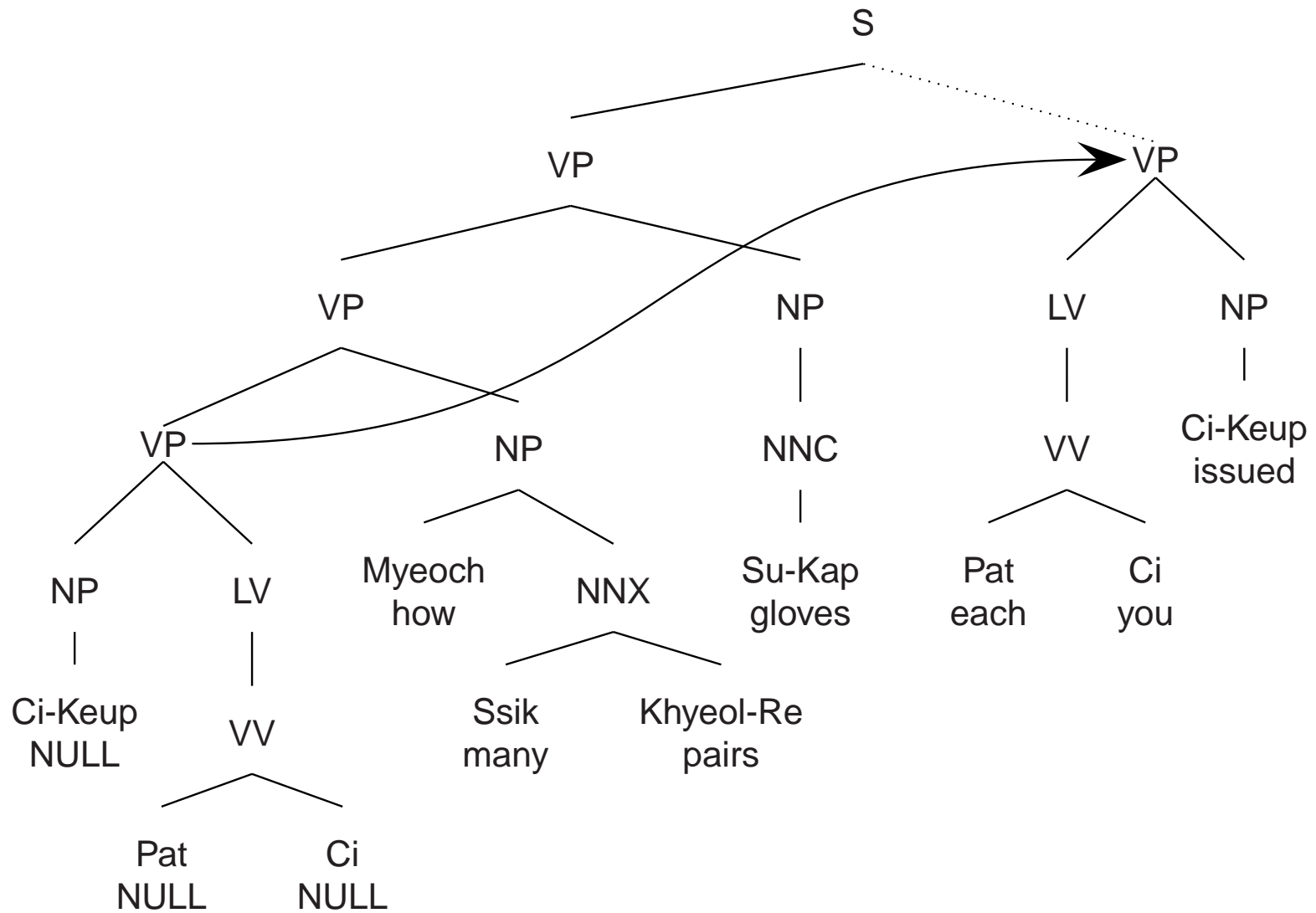
$$P_{ins}(\text{clone}|\varepsilon_j)$$

$$P_{clone}(\varepsilon_i|\text{clone} = 1) = \frac{P_{makeclone}(\varepsilon_i)}{\sum_k P_{makeclone}(\varepsilon_k)}$$

EM Training Procedure

- Compute expected counts for each possible alignment using dynamic programming (E-step)
- For each node and each span in input string
 - Consider all possible reorderings of children
 - Insertions
 - Cloning is extension of insertion
 - $O(|T|n^3m!2^m)$ where T is input tree, n sentence length, m grammar fan-out
- Re-estimate re-order, insert, and translation probabilities, as well as $P_{ins}(\text{clone}|\varepsilon)$ and $P_{makeclone}(\varepsilon)$ (M-step)

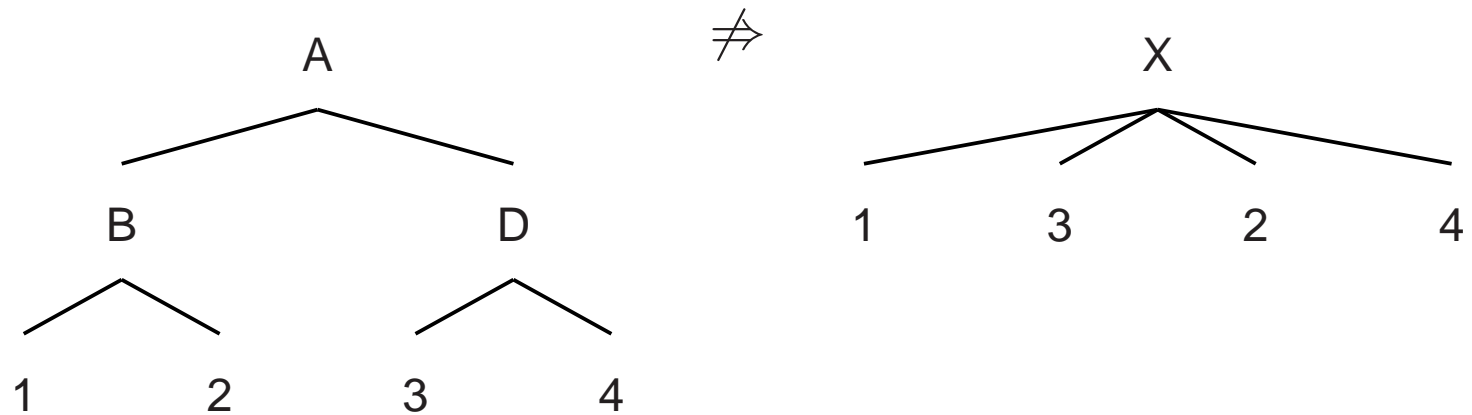
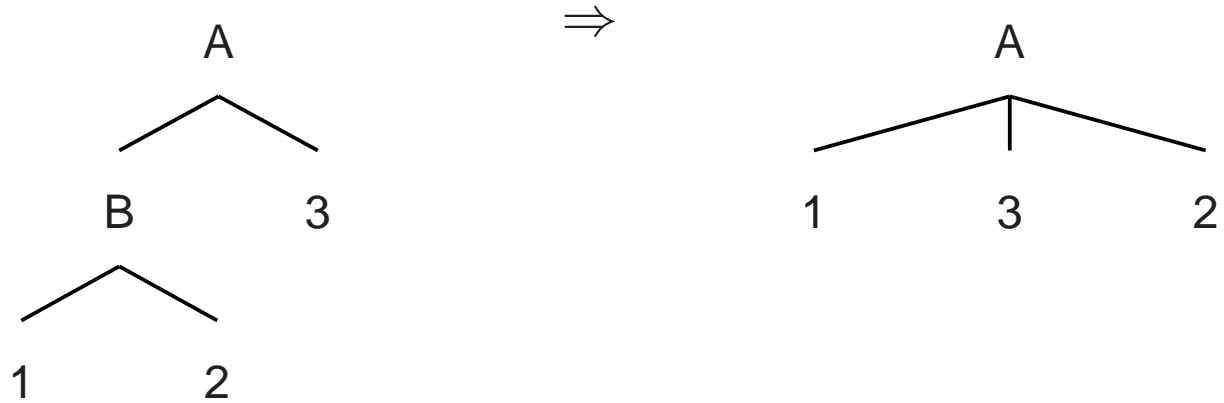
Cloning example



Tree-to-Tree Translation Model

- Result of tree reordering must match tree in target language.
- In addition to previous operations, two-to-one or two-to-two correspondences are allowed between nodes in the two trees:

Elementary Tree Pairs



Tree-to-Tree Parameterization

	Tree-to-String	Tree-to-Tree
tree decomposition		$P_{elem}(t_a \varepsilon_a \Rightarrow children(\varepsilon_a))$
re-order	$P_{order}(\rho \varepsilon \Rightarrow children(\varepsilon))$	$P_{align}(\alpha \varepsilon_a \Rightarrow children(t_a))$
insertion	$P_{ins}(\text{left, right, none} \varepsilon)$	α can include “insertion” symbol
lexical translation	$P_t(f e)$	$P_t(f e)$
with cloning	$P_{ins}(\text{clone} \varepsilon)$ $P_{makeclone}(\varepsilon)$	α can include “clone” symbol $P_{makeclone}(\varepsilon)$

EM Training Procedure

- Dynamic programming over $T_1 \times T_2$
- For each node pair, consider insertions, deletions, and reorderings of children
- Clone is extension of insertion

$$P_{clone}(\varepsilon_i | \text{clone} \in \alpha) = \frac{P_{makeclone}(\varepsilon_i)}{\sum_k P_{makeclone}(\varepsilon_k)}$$

- $O(|T|^2 m^2 4^{2m} (2m)!)$

Korean-English Parallel Treebank

- 5083 sentences, human-translated and treebanked
- broke apart multi-morphemic Korean words according to POS tags
- average of 21 tokens for the Korean sentences
- average English sentence length was 16

Results

	<i>Alignment Error Rate</i>
IBM Model 1	.37
IBM Model 2	.35
IBM Model 3	.43
Tree-to-String	.42
Tree-to-String, Clone	.36
Tree-to-String, Clone $P_{ins} = .5$.32
Tree-to-Tree	.49
Tree-to-Tree, Clone	.36

Test data: 101 human-aligned sentence pairs.

Summary

- Relaxing tree-based model improves alignments, both for tree-to-string and tree-to-tree.
- Tree-to-String outperforms IBM models when tuned to precision/recall tradeoff.

Now working on:

- Scaling code to more data
- Alternative parameterizations of local reordering
- Integration with rescoring