
Algorithms for Syntax-Aware Statistical MT

I. Dan Melamed

Proceedings of the Conference on Theoretical and Methodological
Issues in Machine Translation (TMI'04), Baltimore, MD.

presented by Jeremy G. Kahn

28 April 2005

overview

- SMT and syntax
- translation as a parsing problem
- the Melamed parser
- using monolingual data with the parser
- The future

transformations across bitexts

- Standard SMT model is string-to-string (transducer)
- language has hierarchical/structural information that may be useful to capture in translation

Syntax-based MT

work we've seen before

- Tree-to-Tree, Tree-to-string: Yamada, Knight, Och
- Stochastic inversion transduction grammars (Wu)

what Melamed's getting at: a pair of parsers makes good linguistic sense, but may or may not be practical.

Multitext Grammars

Generalization of CFG

$$\begin{array}{l} X \\ Y \\ Z \end{array} \Longrightarrow \bowtie \begin{array}{l} [1,2] \\ [1] \\ [1,2,1] \end{array} \left(\begin{array}{cc} A & B \\ () & C \\ D(2) & E \end{array} \right) \quad (1)$$

Represents a 3-text, non-terminal rule.

- role templates
- link columns
- join operator
- discontinuities

Multitext Grammars

$$\begin{array}{c} () \\ Y \end{array} \implies \begin{pmatrix} () \\ C \end{pmatrix} \quad (2)$$

Represents a terminal rule in a 2-text. By definition they are always active in only one text.

Translation as probabilistic parse search

Items:

$$\left[\begin{array}{l} X_1 \\ X_2 \end{array} ; \sigma \right]$$

Goal:

$$\left[\begin{array}{l} S_1 \\ S_2 \end{array} ; (0, n) \right]$$

Inference in the parser

Scan:

$$\frac{G_T \left(\begin{array}{c} Y \\ () \end{array} ; \begin{array}{c} w_i \\ () \end{array} \right)}{\left[\begin{array}{c} Y \\ () \end{array} ; \begin{array}{c} (i-1), i \end{array} \right]}$$

Load:

$$\frac{G_T \left(\begin{array}{c} () \\ Z \end{array} ; \begin{array}{c} () \\ t \end{array} \right)}{\left[\begin{array}{c} () \\ Z \end{array} ; \begin{array}{c} () \end{array} \right]}$$

Inference in the parser

Compose:

$$\frac{\left[\begin{array}{c} Y_1 \\ Y_2 \end{array} ; \tau \right] \left[\begin{array}{c} Z_1 \\ Z_2 \end{array} ; \sigma \right] G_n \left(\begin{array}{c} X_1 \\ X_2 \end{array} ; \begin{array}{c} \tau \otimes \sigma \\ \rho \end{array} , \begin{array}{c} Y_1 \\ Y_2 \end{array} , \begin{array}{c} Z_1 \\ Z_2 \end{array} \right)}{\left[\begin{array}{c} X_1 \\ X_2 \end{array} ; \tau + \sigma \right]}$$

- X_1 and X_2 are the parents in languages 1 and 2
- Y_1 , Y_2 , Z_1 , and Z_2 are the children
- $\tau \otimes \sigma$ represents the role template in the input language of the new parent
- ρ is the role template in the output language

Why a general parser?

- leaves out the issue of chart selection and ordering (CKY, Earley, etc)
- abstracts away from Viterbi, k -best, etc
- use different semirings for different behaviors

Synchronous parsers as joint models on paired trees

$$\begin{aligned} G_n \left(\begin{array}{c} X_1 \\ X_2 \end{array} ; \begin{array}{c} \rho_1 \\ \rho_2 \end{array} , \begin{array}{c} Y_1 \\ Y_2 \end{array} , \begin{array}{c} Z_1 \\ Z_2 \end{array} \right) &= \Pr(\rho_1, \rho_2, Y_1, Y_2, Z_1, Z_2 | X_1, X_2) \\ &= \Pr(\rho_2, Y_2, Z_2 | X_1, X_2) \\ &\quad \times \Pr(\rho_1, Y_1, Z_1 | \rho_2, Y_2, Z_2, X_1, X_2) \end{aligned}$$

$\Pr(\rho_2, Y_2, Z_2 | X_1, X_2)$ is “analogous to the language model in noisy-channel decomposition”.

But how to use n -grams or other monolingual models in this context?

Combining with bigram model

- problem: discontinuous generation (even without discontinuous rules)
- naive approach: do eval after generation (causes trouble with pruning, also hard to use this to direct search)
- better A*-style approach: use n -gram if possible to direct search
- thus, need to be able to compute probability of substrings

Clever, clever

- treat bigram as binary-expansion right-branching tree
 - can build up from edges

$$\frac{BG(s_1.s_2) = BG(s_1) \times BG(s_2) \times BG(s_2[0]|s_1[-1])}{BG(end|s_1[-1]) \times BG(s_2[0]|begin)}$$

- in log space, this is sum bigram probs of substring regions plus the bigram across the gap, minus the bigram probs of the start and end tokens for each
- generalizes for multiple fused pairs (eq 14)
- revised parser (fig 4 in paper) stores edge info for bigram conditioning computation at each step

Compose inference revised to use bigram

now bigram can be used to compute the adjacency constraints on the output language (eq 16)

$$G_n \left(\begin{array}{c} X_1 \\ X_2 \end{array} ; \begin{array}{c} \rho_1 \\ \rho_2 \end{array} , \begin{array}{c} Y_1 \\ Y_2[\lambda] \end{array} , \begin{array}{c} Z_1 \\ Z_2[\mu] \end{array} \right) = BG(\%(\rho_2, \lambda, \mu)) \\ \times TM(\rho_1, Y_1, Z_1 | \rho_2, Y_2, Z_2, X_1, X_2)$$

Independence assumptions:

$$\Pr(\rho_2, Y_2[\lambda], Z_2[\mu] | X_1, X_2) = BG(\%(\rho_2, \lambda, \mu)) \\ \Pr(\rho_1, Y_1, Z_1 | \rho_2, Y_2, Z_2, X_1, X_2) = TM(\rho_1, Y_1, Z_1 | \rho_2, Y_2, Z_2, X_1, X_2)$$

Related application: evaluation

train a probabilistic synchronous CFG over a *monolingual multitext*

- allowing certain syntactic swaps should be okay
- synchronous CFGs could allow these parallel tokens to be the same
- probabilistic synchronous CFG could score the *likelihood* of these being the same (instead of translation model)
- extensions of this idea: paraphrasing, (summarization?)

What's next?

- Grammars
 - how to train the “translation model”?
 - MT CFG kinda clumsy
 - add inflectional morph?
 - add non-compositional compounding?
- Logic for parser
 - CKY is presumably fine, but Earley or other? search order?
- semiring objectives
 - how to decide which to use?
 - Max Ent is current proposal, but margin methods?
- O Search: Melamed claims high-polynomial time for this algorithm (claims low-exponential for other syntax SMT).

Discussion

- construction of PCFGs from monolingual data is not completely solved
- synchronization of PCFGs is also fairly challenging
- given parallel treebanks, would this be easier or harder?
- this is a search over trees. BLEU, and humans, search over words. Do we care that non-optimal tree spans may generate the same words?

Melamed has a “forthcoming” in *Computational Linguistics* on these questions.