A Simple Refinement to Cube Pruning for Syntax-Based Statistical Machine Translation

Wenduan Xu[1] and Philipp Koehn[2]
[1]Computer Laboratory, The University of Cambridge
[2]School of Informatics, The University of Edinburgh
wenduan.xu@cl.cam.ac.uk, pkoehn@inf.ed.ac.uk

Introduction

- Syntax-Based Statistical Machine Translation
  - Incorporating syntactic structures into statistically-oriented MT models
  - Provided promising translation quality gains for many language pairs
  - However, phrase-based translation models still dominate most language pairs in terms of decoding speed due to their simplicity compared with syntactic models
- Decoding Complexity
  - The complexity of syntax-based models introduce additional computational costs into parameter estimation (training) and translation search (decoding)
  - Exact dynamic programming computationally intractable due to exponential dynamic programming states
  - Frequent queries of large *n*-gram language models also introduce additional decoding runtime overhead
- Decoding with a Language Model
  - Induces a lexically exploded dynamic program where each state $[X, i, j]$ is further augmented with two strings of length *n* - 1, composed of the left and right boundary words of a translation hypothesis (n is the language model order)
  - Thus a language model context augmented state is represented as $[X, i, j, l_{1...n-1}, r_{k-n+2...k}]$ where k is the length of the translation hypothesis, $l_{1...n-1}$ and $r_{k-n+2...k}$ are the left and right n- 1 boundary words of that hypothesis, respectively
  - Decoding becomes practically infeasible with $\mathcal{O}\left(m^{3+4(n-1)}\right)$ complexity

Contributions

- We present a simple refinement of cube pruning based on a first full inside-outside parsing pass to generate inside and outside cost products to augment the second pass +LM decoding
- We demonstrate the effectiveness of our approach on a Chinese-English translation task with a hierarchical model and a English-German task with a string-to-tree model
- Boost decoding speed by 20% and 38% on average for two translation models, respectively, without compromising translation quality as measured by BLEU

Translation Models

- Model
  - Both translation models we consider are based on SCFGs as in (Chiang, 2007)

$$X \rightarrow \langle \gamma, \alpha, \sim \rangle$$

  - Example rule

$$X \rightarrow \langle X_1 \text{ xiede } X_2,\ X_2 \text{ written by } X_1 \rangle$$

- Rule Restrictions
  - To reduce the grammar size for the target-syntax model
  - Impose restrictions to constituent target phrases by allowing up to seven source-side terminal/nonterminal symbols and discard rules with scope greater than three (Hopkins and Langmead, 2010)
  - These restrictions and the addition of linguistic labels on the target side reduces the total grammar size
  - Also reduces the problem of spurious ambiguity

Decoding

- Decoding without a Language Model
  - Decoding with only a SCFG-based translation model is isomorphic to monolingual bottom-up CKY parsing requiring an $\mathcal{O}\left(m^3\right)$ parsing algorithm
  - A dynamic programming state is identified by its target side nonterminal symbol and the input sentence span covered by it, e.g., $[X, i, j]$
- Cube Pruning
  - Originated from *k*-best parsing algorithms in Huang and Chiang, 2005, and applied to machine translation first in (Chiang, 2007)
  - A heuristic algorithm used to speed up MT decoding with integrated language models
  - The state-of-the-art algorithm which enables approximate dynamic programming and lazy language model querying throughout language model integrated decoding and achieves comparable translation quality as other non-lazy methods

Cube Pruning

- The Cube Pruning Algorithm (Chiang, 2007)

```
1:  procedure MainLoop (H = ⟨V, E⟩)
2:  for X ∈ V in topological order do
3:      SelectK (X, k)

4:  procedure SelectK (X, k)
5:  PriorityQueue ← {h_e (1) | e ∈ BS (X)}
6:  H_top-k ← ∅
7:  PriorityQueue-temp ← ∅
8:  while |PriorityQueue-temp| < k and
          |PriorityQueue| > 0 do
9:      h_e ← PriorityQueue.pop-min_≤
10:     PriorityQueue-temp.push (h_e)
11:     for h'_e ∈ CreateNeighbours (h_e (u)) do
12:         if h'_e ∉ PriorityQueue then
13:             PriorityQueue.push (h'_e)
14: H_top-k ← PriorityQueue-temp.pop-all.sort

15: procedure CreateNeighbours (h_e (u))
16: N ← ∅
17: for i ← 1 ... |e| do
18:     h'_e ← h_e (u + b_i)
19:     if (u + b_i)_i ≤ |H_i| then
20:         N.insert (h'_e)
21: return N
```

- The pop-limit variable k on line 8 of the pseudocode controls the runtime cost of cube pruning
- It is a constant applied to every chart cell during decoding and potentially wasting decoding efforts for chart cells for which lower pop limits would suffice, since the search space is inherently nonuniform
- To exploit this nonuniformity, we propose to dynamically adjust the pop limit based on inside and outside cost estimates of target side nonterminals

Generalized Inside-Outside Semi-ring Parsing

- We use generalized inside and outside algorithms in the first pass -LM decoding

```
1: procedure InsideParse (H = ⟨V, E⟩)
2: for X ∈ V in topological order do
3:     for each incoming hyperedge e of X do
4:         for each antecedent node X_i of X do
5:             ω ← ω · β (X_i)
6:         β (X) ← max (β (X), β (X_i) · ω · R_e)
```

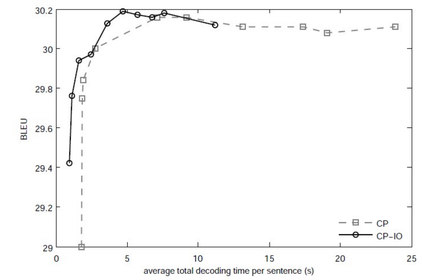
```
1: procedure OutsideParse (H = ⟨V, E⟩)
2: for X ∈ V do
3:     α (X) ← 0

4: for X ∈ V in reverse topological order do
5:     for each incoming hyperedge e of X do
6:         for each antecedent node X_i of X do
7:             α (X_i) ← max(α (X_i), α (X) R_e ∏_{i≠j} β (X_j))
```

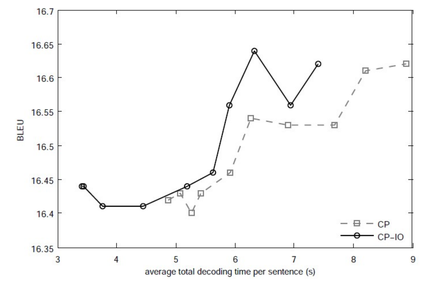
- In the second pass, cube pruning pop-limit parameter is augmented with

$$\mu = \alpha_X [i, j] \beta_X [i, j]$$

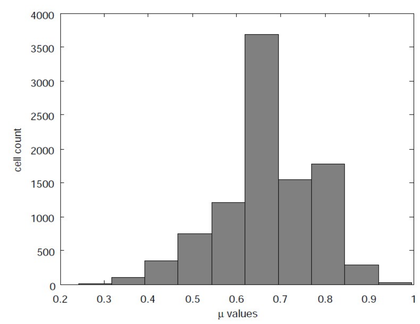
Experiments

- Decoding efficiency and translation quality comparisons on two large scale experiments, NIST08 Chinese-to-English test set (1357 sentences) and WMT10 newstest2009 test set (1004 sentences)





- Example μ values for the Hiero model



References

- D. Chiang. 2007. Hierarchical phrase-based translation. Computational Linguistics, 33(2)
- M. Hopkins and G. Langmead. 2010. Scfg decoding without binarization. In Proc. EMNLP
- L. Huang and D. Chiang. 2005. Better k-best parsing. InProc. IWPT