Hierarchical Phrase-based Translation with Weighted Finite-State Transducers

Gonzalo Iglesias
Machine Intelligence Laboratory
Cambridge University Engineering Department

Leuven, 31st of May 2011
Exactness in Search I

- The search space is the intersection of $s^n_1$ with the grammar.
- In order to improve performance we may consider exploring bigger search spaces.
- But if it is too big for the decoder, local pruning is needed.
Exactness in Search II

- Local pruning leads to search errors
- CUED Machine Translation group: Exactness and WFSTs.
  - Phrase-based Translation: TTM
  - HiFST + shallow-$N$ grammars

---

This talk

- Hierarchical Phrase-based Translation
- HiFST: Hiero decoder using WFSTs
- Shallow-$N$ grammars
- Results on several Translation Tasks
  - Contrast with Hiero Cube pruning decoder for AREN and ZHEN
  - HiFST and LMBR (rescoring and system combination)
  - WMT10 (FR/EN and SP/EN tasks)
Hierarchical Phrase-based Translation

- Hierarchical grammars and decoders first introduced by Chiang\(^2\)
- Hierarchical rules (phrases with gaps) allow generalization and reordering
- These rules are formulated as a synchronous grammar

\[
\begin{align*}
S & \rightarrow \langle X, X \rangle \\
S & \rightarrow \langle S \ X, S \ X \rangle \\
X & \rightarrow \langle \gamma, \alpha, \sim \rangle, \gamma, \alpha \in \{X \cup T\}^+ \\
\end{align*}
\]

| \(S \rightarrow \langle X, X \rangle\) | glue rule 1 |
| \(S \rightarrow \langle S \ X, S \ X \rangle\) | glue rule 2 |
| \(X \rightarrow \langle \gamma, \alpha, \sim \rangle\) | hiero rules |

Table: Rules contained in the standard hierarchical grammar.

- Gaps have no syntactic meaning
- Greedy automatic extraction from aligned parallel data, with standard constraints\(^3\)
- For search: Hierarchical Cube pruning decoders,...


HiFST

- A hierarchical decoder that uses lattices\textsuperscript{4} for translation hypotheses.
- Why use lattices?
  - Compactness
  - Easily represented with Weighted Finite-State Transducers (WFSTs).
  - WFSTs: OpenFST, available at openfst.org\textsuperscript{5}
  - Weights typically represented as costs under Tropical Semiring
    \[
    \{ R, \oplus = \text{min}, \otimes = +, 0 = \infty, 1 = 0 \}
    \]
  - Standard WFST Operations templated over semiring (minimize, compose, prune shortestpath, ...) handle efficiently weights
  - Key for success: RTNs
- WFSTs: successfully applied to several NLP tasks


\textsuperscript{5}Allauzen, Cyril, Michael Riley, Johan Schalkwyk, Wojciech Skut, and Mehryar Mohri. OpenFst: A general and efficient weighted finite-state transducer library. CIAA, 2007.
HiFST - General Framework

- CYK algorithm: source side
  - Given a sentence $s_1 \ldots s_J$, and a synchronous grammar, find all derivations with root in cell $(S, 1, J)$
- Lattices $\mathcal{L}(N, x, y)$ are built for each cell following back-pointers of the grid
  - Objective is the expanded lattice $\mathcal{L}(S, 1, J)$, at the top of the grid
- Apply language model to $\mathcal{L}(S, 1, J)$ and prune
**Lattice Building**

```plaintext
1 function buildFst(N,x,y)
2   if \( \mathcal{L}(N, x, y) \) exists, return \( \mathcal{L}(N, x, y) \)
3   for each rule applied in cell \((N, x, y)\),
4     for each element in rule
5       if element is a word, create \( A(\text{element}) \)
6     else buildFst(backpointers(\text{element}))
7   Create rule lattice by catenation of element lattices
8   Create cell lattice \( \mathcal{L}(N, x, y) \) by unioning rule lattices
9   Reduce \( \mathcal{L}(N, x, y) \) with FST operations and return

- Recursive algorithm with memoization – traverses grid and returns RTN for the topmost cell. Lower level lattices also stored\(^6\).
```

Our lattices are a mixture of words and pointers to lower level lattices (RTNs)

Topmost cell lattice $\mathcal{L}(S, 1, J)$ is expanded.
Lattice Expansion II

- Usual operations (rmepsilon, determinize, minimize, etc) work over RTNs (and keep them small)!
Shallow-$N$ grammars I

- $R^1$: $S \rightarrow \langle X, X \rangle$
- $R^2$: $S \rightarrow \langle S \ X, S \ X \rangle$
- $R^3$: $X \rightarrow \langle X \ s_3, t_5 \ X \rangle$
- $R^4$: $X \rightarrow \langle X \ s_4, t_3 \ X \rangle$
- $R^5$: $X \rightarrow \langle s_1 \ s_2, t_1 \ t_2 \rangle$
- $R^6$: $X \rightarrow \langle s_4, t_7 \rangle$

- For certain language pairs, this rule nesting might be unnecessary
Shallow-$N$ grammars II

- Allowing only one level of hierarchical rule nesting is trivial:
  
  $R^1: \ S \rightarrow \langle X, X \rangle$
  $R^2: \ S \rightarrow \langle S \ X, S \ X \rangle$
  $R^3: \ X_1 \rightarrow \langle X_0 \ s_3, t_5 \ X_0 \rangle$
  $R^4: \ X_1 \rightarrow \langle X_0 \ s_4, t_3 \ X_0 \rangle$
  $R^5: \ X_0 \rightarrow \langle s_1 \ s_2, t_1 \ t_2 \rangle$
  $R^6: \ X_0 \rightarrow \langle s_4, t_7 \rangle$

- Easily extended to any $N$ levels: Shallow-$N$ grammars.

- Limiting rule nesting to a fixed threshold is a kind of derivation filtering.
Performance

- Lattice output has benefits for lattice rescoring and system combination:
  - Large Language Model rescoring
  - Lattice MBR for rescoring and system combination\(^7\) \(^8\)

- Translation tasks between close languages do not require complex rule nesting – Shallow-1 grammars reach similar state-of-the-art performance with much faster decoding times
  - Arabic-to-English\(^9\)
  - Spanish-to-English\(^10\)

\(^9\) See EACL 2009 paper.
Very competitive translation systems using shallow-1 grammars\textsuperscript{11}

<table>
<thead>
<tr>
<th></th>
<th>news-test2008</th>
<th>newstest2009</th>
<th>newstest2010</th>
</tr>
</thead>
<tbody>
<tr>
<td>SP-EN</td>
<td>25.4</td>
<td>27.0</td>
<td>30.5</td>
</tr>
<tr>
<td>EN-SP</td>
<td>24.7</td>
<td>25.5</td>
<td>29.1</td>
</tr>
<tr>
<td>FR-EN</td>
<td>25.6</td>
<td>29.3</td>
<td>29.6</td>
</tr>
<tr>
<td>EN-FR</td>
<td>24.2</td>
<td>26.1</td>
<td>28.2</td>
</tr>
</tbody>
</table>

\textbf{Table:} WMT10 HiFST+LMBR Translation Systems

Shallow-$N$ for ZHEN

- Chinese-to-English task: word reordering requirements are harsh.
- Shallow-3 almost bridges the gap at faster decoding times
HiFST versus Cube Pruning I

- HiFST with cube pruning decoder for Arabic-to-English translation task (shallow-1). No search errors + lattices = increased performance.
HiFST vs Cube Pruning II

ZHEN task vs. local pruning

- We still have local pruning on ZHEN translation
- Full grammars define search spaces far too big
- Is it possible to avoid local pruning for ZHEN translation tasks?
- Yes: See our next paper in EMNLP 2011\textsuperscript{12}! Recipe:
  - Push-down automata
  - 1st pass with entropy pruned language model
  - Rescoring with full language model

HiFST goes online – FAUST project

- FAUST: Feedback Analysis for User Adaptive Statistical Translation
- Motivation: Current MT systems do not respond to suggestions for improvement. There are diverse technical reasons for this, including:
  - User feedback tends to be very noisy
  - No research published to date makes explicit how statistical translation and language models can be adapted to benefit from feedback provided by web users
  - No mechanisms exist to identify user feedback of value and immediately change behaviour of SMT systems in order to avoid the problem
  - Current SMT systems and research efforts are aimed at sophisticated users - translation professionals, intelligence analysts, etc. These users develop an understanding of how to work around their system weaknesses
  - Casual users are tend to be frustrated by a general lack of fluency
HiFST goes online – FAUST project

HiFST will be available very soon. Check www.reverso.net!
Conclusions

- We described HiFST, a hierarchical decoder based on WFSTs
  - Easy to implement, as complexity is hidden by OpenFST library
  - RTNs effectively reduce complexity during lattice construction
- For ZHEN: Shallow-3 almost bridges the gap with Hiero Full
- ZHEN: bigger search spaces – we still need local pruning
- For languages not in need of strong word reorderings, shallow-1 grammars generally enough
- AREN and WMT10: no search errors: exact decoding
Thank you!

Questions?
System Combination of HiFST systems

- Arabic-to-English NIST09 MT08.
- Three HiFST systems over same Arabic sources with different tokenizations (MADA, SAHKR)
- LMBR combines word lattices and searches for hypotheses with highest similarity to the rest of the lattice
Marginalization over Translation Derivations

- Pruned lattices mapped to log semiring – determinization leads to improved performance
- Improvements do not carry through after LMBR step