The CUED OpenMT12 Arabic-English and Chinese-English SMT Systems

Bill Byrne

Adrià de Gispert, Gonzalo Iglesias, Juan Pino, Rory Waite

Department of Engineering
UCAM SMT Systems - Overview

- Lattice-based Hierarchical SMT system implemented with WFSTs
  - HiFST decoder -- based on the Google OpenFST toolkit
- Alignments with the MTTK HMM toolkit
  - All allowable parallel text
- Hybrid systems for Ar-EN and Zh-En
  - Multiple source-side segmentations
  - Genre-specific lexical and translation features
  - MERT & Pro tuning
- Zero-cutoff stupid-backoff 5-gram LMs
  - Google n-gram data
- Lattice Minimum Bayes Risk Decoding
Source Language Text Processing

- Zh-En: 3 alternative segmentations
  - Stanford PKU and CTB [1]
  - Joint Word Segmenter / POS-tagger trained for low OOVs [2]
- Ar-En: 2 alternative segmentations
  - Both provided by MADA [3]

Word-to-Phrase HMM models used for alignment [4]


Hiero Translation Grammars

- Separate grammar extracted for each source segmentation
  - Ar-En: Shallow-1 grammar with long-distance verb movement [5]
  - Zh-En: Full Hiero grammars
- Hadoop/Hfile framework for grammar extraction and retrieval
- Some monotonic rules excluded by pattern
  - Zh-EN: e.g. <X1 w X2, w X1 w X2>, <X w, X w>
  - Ar-En: <w X, w X>
- 10 instances for hierarchical rules, and rule filtering by probability

Features and Feature Weight Tuning

• The usual collection of translation rule features (next slide)

• Provenance Features
  • Translation rule probabilities and lexical features estimated on genre-specific portions of the parallel texts
    • Essentially one translation grammar for each portion of the parallel text

• Feature weights optimized towards BLEU with MERT [6] and Pro [7]
  • Pro was particularly good with sparse features, but favored shorter hyps
  • Translation length models were used to tune towards longer hypotheses


<table>
<thead>
<tr>
<th>Description</th>
<th># AR-EN</th>
<th># ZH-EN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source-to-target probability</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Target-to-source probability</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Word penalty</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Rule Penalty</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Rule Count = 1,2,&gt;3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Deletion and OOV rule</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Is Glue ?</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Source-to-target lexical probability</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Target-to-source lexical probability</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Provenance source-to-target probability</td>
<td>31</td>
<td>9</td>
</tr>
<tr>
<td>Provenance target-to-source probability</td>
<td>31</td>
<td>9</td>
</tr>
<tr>
<td>Provenance source-to-target lexical probability</td>
<td>31</td>
<td>9</td>
</tr>
<tr>
<td>Provenance target-to-source lexical probability</td>
<td>31</td>
<td>9</td>
</tr>
<tr>
<td>Is source length = 1..7 ?</td>
<td>7</td>
<td>Not used</td>
</tr>
<tr>
<td>Is target length = 1..7 ?</td>
<td>7</td>
<td>Not used</td>
</tr>
<tr>
<td>Is source length = 1..7 and target length = 1..7 ?</td>
<td>49</td>
<td>Not used</td>
</tr>
<tr>
<td>Does rule have pattern P ?</td>
<td>65</td>
<td>Not used</td>
</tr>
</tbody>
</table>

- **AR-EN**: provenance features split by genre and LDC collection ID
  - genres: un, nw, ng, bn, bc, wl, treebank

- **ZH-EN**: provenance features split by genre:
  - genres: bc, bn, ng, nw, lexicon, treebank, web, un, null
English Language Models

• LMs used all allowable English text

• English language model is the equal interpolation of
  • KN smoothed 4-gram estimated over the parallel text excluding UN data and monolingual data from the English Gigaword Fourth Edition
  • Zero-cutoff stupid-backoff 5-gram LM estimated over all text [8]
    • Google n-grams were included -- helpful for web text
    • Hadoop/Hfile used for n-gram count retrieval & extraction

HiFST Decoder / LMBR decoding

- Each grammar (for each source segmentation) was used to generate a separate translation lattice (with the common English LM)
- Lattice MBR [8] can be used to merge hypothesis from multiple analyses
  - Each lattice represents a posterior distribution over translation hypotheses for that translation grammar
  - Posteriors are interpolated efficiently via WFST operations [9]

HiFST as a generator of statistics for LMBR

- LMBR consistently gives +0.5-1.0 BLEU over shortest-path hypothesis
  - Gains are even greater for lattices from multiple grammars
  - Why: Lattice posteriors can predict translation quality of n-grams in hypotheses
What didn’t work (yet):
Challenges in using more powerful grammars

- HiFST works very well for grammars with relatively few non-terminals
- Richer grammars such as SAMT, GHKM, make it difficult to determinise the hypothesis space during search, which is crucial for good performance
- Alternative decoder architectures are showing some promise [10]

Summary

- Flexible grammar configuration strategies
  - Full Hiero for Zh-En
  - Shallow-N for Ar-En
  - Many features, including provenance
- Hybrid translation systems
  - Good gains from lattice MBR based on multiple source language analyses
    - Adds robustness for noisy source text
- Emphasis on efficient construction of translation search spaces
  - Minimal pruning
  - Goal: few search errors
- Lattice MBR provides a straightforward and robust method to combine hypotheses from multiple translation grammars
Acknowledgements

• Graeme Blackwood: initial Pro implementation and the LMBR tools
• Stanford CoreNLP suite, SRI LM Tools, MADA
• The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7-ICT-2009-4) under grant agreement number 247762 (FAUST project)
• R. Waite, J. Pino supported by EPSRC(UK) Doctoral Training Awards
• DARPA GALE and BOLT programs

Department of Engineering
University of Cambridge