Overview

- Hierarchical phrase-based system using cdec
- Constrained track
- Gains through source-side reordering, domain adaptation, large and class-based language models (LMs)
- Large-scale tuning with sparse, lexicalized features
- \textit{K}-best rescoring with syntax-based and neural network LMs

Training pipeline

Source-side reordering \quad We re-arranged all source-sentences to match the syntax of the target language by applying a variation of the approach described in [Genzel, 2010] \quad \rightarrow +0.1–0.37 BLEU

Domain adaptation \quad We added a 4-gram in-domain language model and annotated each hierarchical phrase with indicators for each training corpus, allowing the model to learn log-linear scaling weights for each corpus. \quad \rightarrow +0.3 BLEU

Alignment indicator features \quad We included lexicalized alignment indicator features which model word alignment, deletion and insertion in source and target. \quad \rightarrow +0.16–0.29 BLEU

Larger language models \quad We built a 5-gram word-based language model, and a 7-gram class-based language model (c=200) from 26.8 million German sentences including the training data target side, News Crawl and political speeches. \quad \rightarrow +1.4–2 BLEU

\texttt{GIZA++} \quad Our experiments confirmed that training alignments with \texttt{GIZA++} gave a significant boost in performance. \quad \rightarrow +1.01–1.6 BLEU

Figure 1: Components of the training pipeline.

Large-scale tuning

- Wide range of sparse features, tuned on three development sets:
  - \textit{rule identity features (id)} \quad one binary feature per rule.
  - \textit{rule shape features (shape)} \quad generalized rules, by mapping sequences of terminal and non-terminals to place holders and word classes.
  - \textit{rule bigram features (bigram)} \quad all bigrams of terminals and non-terminals inside rules, in both source and target side.

- We employ an online variant of pairwise ranking optimization with data sharding and feature selection by $\ell_1$/$\ell_2$ regularization and randomization of the training input.
- Sharding of the data greatly improves efficiency, as the tuning and optimization may run on several parts of the data at once.
- Models of different shards and training iterations are mixed via averaging.

Data & baseline system

- Data preprocessing: Filter sentences longer than 150 words, filter wrong languages from Common Crawl, tokenize, truecase.

- Baseline model: 21 features (4 bidirectional phrase and word pair probabilities, 7 pass-through features, 3 arity penalty features, a 4-gram target side LM, count features for word penalty, glue rules, and language model OOVs), tuned on IWSLT dev2010.

Software

- \texttt{otedama} (automatic preordering): github.com/StatNLP/otedama
- \texttt{dtrain} (parallel pairwise ranking): github.com/pks/cdec-dtrain
- \texttt{cdec} (decoder): github.com/redpony/cdec

$k$-best rescoring

- We incorporated more knowledge sources via $k$-best rescoring (k=100): 3 in-domain language models built from part-of-speech, morphology and lemma annotation. In-domain and a target-side feed-forward neural network LMs using the NPLM toolkit (nlp.isi.edu/software/nplm/).

- Weights for the different language models were learned using a PRO-style pairwise ranking approach, with an SGD classifier from scikit-learn.

- Rescoring achieved no BLEU-gains over the large-scale system, but was preferred in 62 percent of the cases in a small human pairwise preference evaluation.

Final results

<table>
<thead>
<tr>
<th></th>
<th>tst2014</th>
<th>tst2015</th>
</tr>
</thead>
<tbody>
<tr>
<td>Official Baselines</td>
<td>18.49</td>
<td>20.08</td>
</tr>
<tr>
<td>Contrastive (large-scale, no rescoring)</td>
<td>23.24</td>
<td>25.22</td>
</tr>
<tr>
<td>Primary (large-scale + rescoring)</td>
<td>23.22</td>
<td>24.96</td>
</tr>
</tbody>
</table>

Figure 2: Ablation test for sparse features on tst2013.