The Heidelberg University English-German translation system for IWSLT 2015

Laura Jehl, Patrick Simianer, Julian Hitschler and Stefan Riezler

Department of Computational Linguistics, Heidelberg University

UNIVERSITÄT HEIDELBERG
ZUKUNFT SEIT 1386

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
- K-best rescoring with syntax-based and neural network LMs

Training pipeline

Source-side reordering 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]. → +0.1–0.37 BLEU

Domain adaptation 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. → +0.3 BLEU

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

Larger language models 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. → +1.4–2 BLEU

GIZA++ Our experiments confirmed that training alignments with GIZA++ gave a significant boost in performance. → +1.01–1.6 BLEU

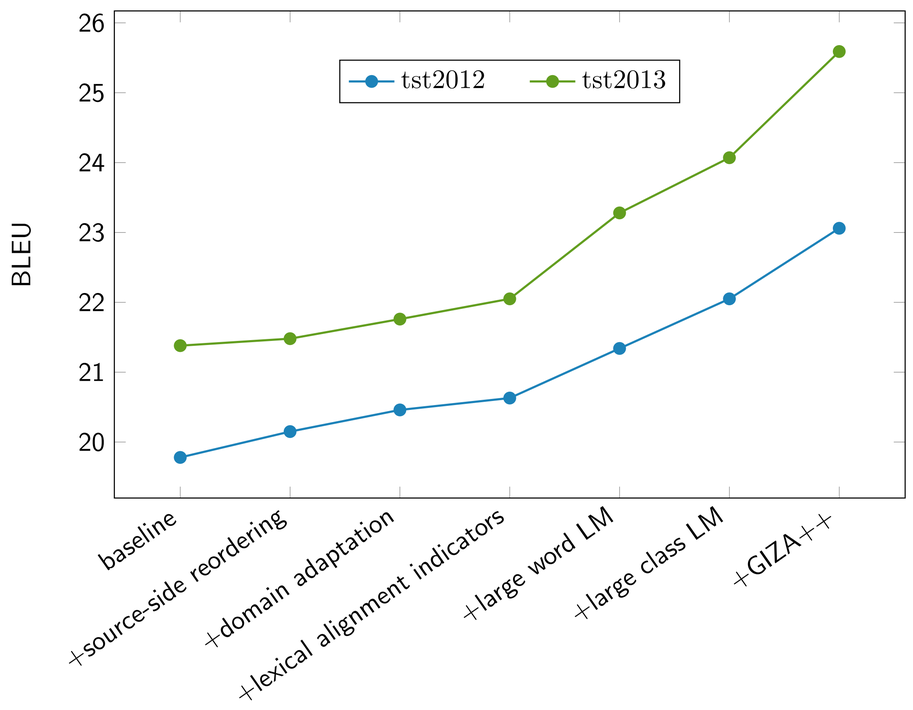

Figure 1: Components of the training pipeline.

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

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

Large-scale tuning

- Wide range of sparse features, tuned on three development sets:

rule identity features (id) one binary feature per rule.

rule shape features (shape) generalized rules, by mapping sequences of terminal and non-terminals to place holders and word classes.

rule bigram features (bigram) 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.

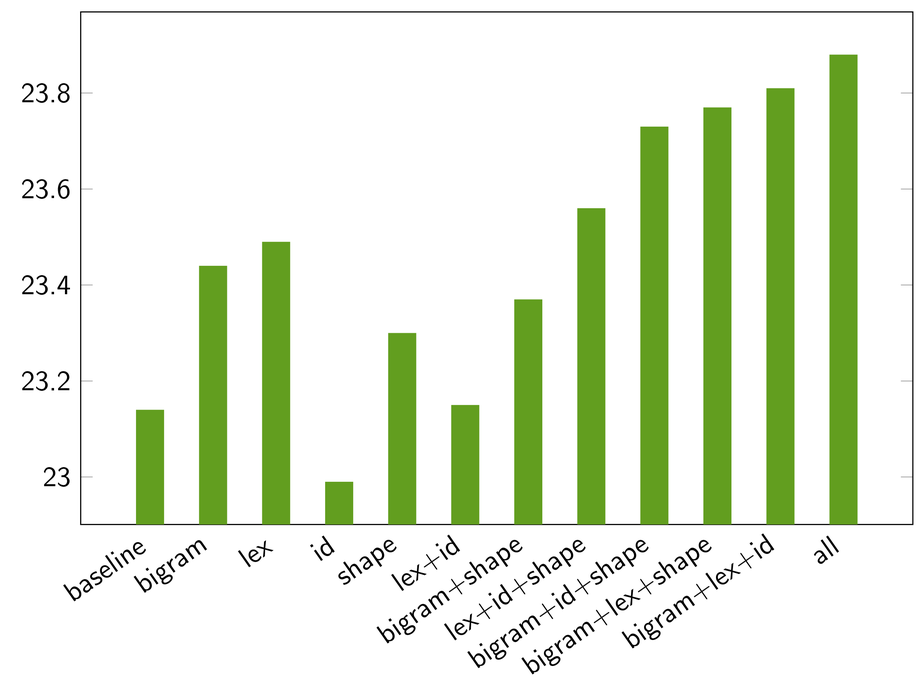

Figure 2: Ablation test for sparse features on tst2013.

- In isolation, rule identifiers, bigram and shape features did not help (much).
- However, combining rule identifiers with other sparse features resulted in improvements – e.g. bigram+id+shape improved by about 0.6 BLEU over the baseline.
- Combining all sparse features worked best.

Final results

	tst2014	tst2015
Official Baselines	18.49	20.08
Contrastive (large-scale, no rescoring)	23.24	25.22
Primary (large-scale + rescoring)	23.22	24.96

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 (`nlg.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.