Stanford University's Arabic-to-English Statistical Machine Translation System for the 2009 NIST MT Open Evaluation

Michel Galley, **Spence Green**, **Daniel Cer**, **Pi-Chuan Chang**, and **Christopher D. Manning**
Computer Science and Linguistics Departments
Stanford University
{mgalley,spenceg,cerd,pichuan,manning}@stanford.edu

Abstract

This document describes Stanford University's first entry into a NIST Arabic-English MT evaluation. We describe two main improvements over a previous Chinese-English submission (Galley et al., 2008): a hierarchical lexicalized reordering model (Galley and Manning, 2008) and a technique for performing minimum error rate training (Cer et al., 2008) that outperforms the standard Powell method.

1 System Description

1.1 Phrase-based translation system

The core engine of our system is Phrasal, a phrase-based decoder similar to Moses (Koehn et al., 2007). In its baseline configuration and basic set of features, Phrasal replicates Moses almost exactly, and differs only in the way the decoder breaks ties between translation hypotheses that have the same score. Unless otherwise indicated, we use the same default parameters as Moses (e.g., same recombination heuristic, same maximum number of translation options for each input phrase).

Phrasal uses a log-linear approach common to many state-of-the-art statistical machine translation (SMT) systems (Och and Ney, 2004). Given an input Arabic sentence **f**, which is to be translated into an English sentence **e**, the decoder searches for the most probable translation **ê** according to the following decision rule:

$$\hat{\mathbf{e}} = \arg\max_{\mathbf{e}} \{P(\mathbf{e}|\mathbf{f})\} = \arg\max_{\mathbf{e}} \{\sum_{m=1}^{M} \lambda_m h_m(\mathbf{f},\mathbf{e})\}$$

where $h_m(\mathbf{f},\mathbf{e})$ are M arbitrary feature functions over sentence pairs, such as translation probabilities. Our system incorporates the following 17 feature functions:

- **Two phrase translation probabilities** $P_{ml}(\overline{e}|\overline{f})$ and $P_{ml}(\overline{f}|\overline{e})$, computed using the (unsmoothed) relative frequency estimate

  $$P_{ml}(\overline{e}|\overline{f}) = \text{count}(\overline{e},\overline{f}) / \big(\sum_{\overline{e'}} \text{count}(\overline{e'},\overline{f})\big),$$

  where $\overline{f}$ and $\overline{e}$ constitute a pair of aligned phrases.

- **Two lexical translation probabilities** $P_{lex}(\overline{e}|\overline{f},\mathbf{a})$ and $P_{lex}(\overline{f}|\overline{e},\mathbf{a})$, similar to those presented in (Koehn et al., 2003):

  $$P_{lex}(\overline{e}|\overline{f},\mathbf{a}) = \prod_{i=1}^{n} \frac{1}{|\{i|(i,j)\in\mathbf{a}\}|} \sum_{(i,j)\in\mathbf{a}} p(\overline{e}_i|\overline{f}_j),$$

  where n is the length of the phrase $\overline{e}$, and **a** is the internal word alignment between $\overline{e}$ and $\overline{f}$.[1]

- **Eight hierarchical lexicalized phrase reordering scores** for each phrase pair. We select from four types of orientations (monotone,

[1] Distinct instances of a given phrase pair ($\overline{e}$,$\overline{f}$) may be observed with different internal alignments. In these cases, we select the most frequent alignment (like Moses but in contrast to (Koehn et al., 2003)). About 0.3% of our phrases have lexical translation probabilities that differ from Moses since our feature extraction implementation breaks ties between alignment counts differently. However, we observe no impact on MT performance.

swap, left discontinuous, and right discontinuous) and model both left-to-right and right-to-left re-orderings. Laplace smoothing with $\lambda = 0.5$ is applied to the lexicalized re-ordering probabilities. More details about this model can be found in (Galley and Manning, 2008).

- **Two language models**, from Gigaword and Google *n*-grams.
- **Word penalty** as in (Koehn et al., 2007).
- **Phrase penalty** as in (Koehn et al., 2007).
- **Linear reordering penalty** as defined in (Koehn et al., 2007).

The weights of these feature functions were set using an improved version of minimum error rate training (MERT) (Och, 2003). Specifically, we used a stochastic method and two regularization strategies that are described in (Cer et al., 2008), which shows that this approach is superior to both Powell's method and the variant of coordinate descent found in the Moses MERT utility. Our system was tuned using MT06 (LDC2007E59). We did not tune different systems for different genres.

The decoder used a distortion limit of 5. Stack size and n-best list sizes were set to 500 (Moses's defaults are respectively 200 and 100, which we found less effective). After decoding, hypotheses are selected using the minimum Bayes risk criterion (Kumar and Byrne, 2004).

1.1.1 Phrase tables

This section describes the computation of phrase translation and lexicalized re-ordering probabilities, which we computed for all observed phrases of up to ten words on either side. We used all the parallel Arabic-English (A-E) training data permissible in the constrained track, including ISI's automatically extracted A-E parallel text (LDC2007T08). This parallel data was pre-processed by IBM Research. Pre-processing of the Arabic side involved two steps: normalization and segmentation. Normalization involved, e.g., removing all diacritics and reverting orthographic changes to stems caused by segmented affixes. Arabic words were segmented according to the Penn Arabic treebank (ATB) standard.[2]

We performed word alignment using a cross-EM word aligner (Liang et al., 2006). For this, we ran two iterations of IBM Model 1 (Brown et al., 1993) and two iterations of the homogeneous HMM model described in (Vogel et al., 1996). To increase the weight of good-quality data, we also used the alignment technique presented by BBN during the 2008 Open MT Evaluation. Like BBN, we found that this technique yields systematic performance gains ranging from 0.2 to 0.4 BLEU point. Finally, we generated a symmetric word alignment from cross-EM Viterbi alignment using the Moses grow heuristic.

To construct phrase tables we used an implementation of phrase-extract (Och, 2002), which, unlike Moses, builds phrase tables directly tailored to specific development and test sets. This considerably reduces the burden of computing normalization counts, since the phrase extraction module can usually fit all relevant phrase pairs into memory (as opposed to, e.g., Moses, which sorts large collections of phrases on disk to compute normalization counts). This enabled us to quickly experiment with many phrase extraction heuristics. On a 6.5M million English word subset of the parallel data, we found that the grow alignment symmetrization heuristic (Koehn et al., 2007) worked best. We pruned phrase tables produced with this heuristic by deleting all phrases that did not satisfy $P_{ml}(\bar{e}|\bar{f}) \geq .0001$. This filtering typically yields phrase tables 2 to 3 times smaller, with little observable impact on MT performance (0.1-0.2% BLEU reduction in the worst case). Since we filter the language models against the target side of our phrase tables, deleting very unlikely translations allowed us to considerably reduce *n*-gram count thresholds—i.e., the number of times each *n*-gram must be observed to be included in the language model—and to incidentally capitalize more on *n*-grams that are likely to be seen at decoding time.

1.1.2 Language models

Our system uses two language models built using the SRI language modeling toolkit (SRILM) (Stolcke, 2002). The first model was trained using sto-

[2] LDC A-E catalog numbers LDC2008E61 (p1), LDC2008E62 (p2), and LDC2008E22 (p3).

ries from Xinhua News and AFP, as well as the entire target-language side of the parallel data (Section 1.1.1), which represent a total of about 824 million English tokens, including punctuation. We built a back-off 5-gram language model smoothed with the modified Kneser-Ney algorithm (Chen and Goodman, 1996). Due to memory constraints, we discarded all 4-grams and 5-grams that occurred less than three times.

We built a second language model using Google *n*-grams. Since the Google collection does not contain *n*-grams with counts lower than 40, it is impractical to utilize smoothing techniques (such as Good-Turning or Kneser-Ney) that rely on "counts-of-counts" statistics to estimate the probability of rare events. We relied instead on Jelinek-Mercer smoothing (Bahl et al., 1983) (known as a "count-based" language model in SRILM), which implements a mixture of count-based maximum-likelihood estimators. In our experiments, the *n*-grams of each order were partitioned by counts into 15 buckets (each bearing a unique interpolation weight), and maximum-likelihood estimates typically converged after 3 to 5 iterations of expectation-maximization (EM) (Dempster et al., 1977). Since SRILM falls short of explicitly enumerating all *n*-grams of count-based language models—the model only contains a few distinct interpolation weights—we converted our count-based language model into the kind of back-off language model expected by our decoder (an ARPA file).[3] Since building a back-off language model requires loading all *n*-grams at once into memory, we limited our use to *n*-grams up to order 3 and removed trigrams that appeared less than 200 times in the Google collection.

1.2 Post-Processing

We trained a linear chain conditional random field (CRF) truecaser that uses various NER features described in (Finkel et al., 2005), among them current and contextual words, word-shape features, etc. This truecaser makes four types of predictions: all lowercase (LC), first letter uppercase (UC), all letters uppercase (CA), and mixed case word (MC) (cf. (Lita et al., 2003)). The truecaser was trained on the target side of the parallel data, though we had to exclude the UN Arabic English parallel text (LDC2004E13) due to memory constraints imposed by our very large feature set. As it is typically done with CRFs, our truecaser was trained to maximize the conditional log-likelihood of the training data. Since this objective function may not necessarily correlate with BLEU, we manually adjusted class priors after training in order to maximize BLEU on MT06 (this is similar to the technique described in (Minkov et al., 2006), though Minkov et al. optimized F-measure instead of BLEU). After running this truecaser, we applied two different post-processing steps. First, we disambiguate the mixed case words by looking up a list we extracted from a larger set of training data. Second, we made the first non-punctuation word of every sentence in the UC category.

We also trained an LM truecaser, which uses the scores of a 5-gram case-sensitive language model as its sole feature. Since this truecaser is much more scalable, we could train it with much more data: the entire target side of the parallel data, plus the AFP and Xinhua sections of the Gigaword corpus, 3rd edition. Both truecasers take as input the single best hypothesis generated by our decoder. We finally trained a product-of-experts truecaser (a logistic classifier), which combines the predictions of the CRF and LM truecasers.

Finally, we applied a rule-based post-processing module to transform dates and numbers, capitalize headlines (if applicable), and transform any British English spelling into American English.

1.3 Contrastive System

Our single contrastive submission was trained and tested using a discriminative lexicalized reordering model (Zens and Ney, 2006), which came in addition to our hierarchical reordering model. Since this extra component caused a small decrease in performance on a blind test set (see Section 2), we did not incorporate it into our primary submission.

[3] One way to achieve this is to create an intermediate back-off ARPA language model containing all *n*-grams of interest (e.g., those that may be applicable at decoding time), then rescore this model with our count-based language model using `ngram -rescore-ngram` in SRILM.

Reordering Models	BLEU[%]
Moses	43.79
phrase-based	43.97
hierarchical	44.22

Table 1: Comparison of lexicalized reordering models.

Language Models	BLEU[%]
Gigaword	43.57
Gigaword + Google	44.50

Table 2: Performance with and without a Google LM.

2 Results

We report experimental results using MT08 (LDC2009E08) as a blind test set. All scores are cased BLEU (Papineni et al., 2001) as computed by the official scoring script (`mteval-v13.pl`).

Table 1 compares three types of lexicalized reordering models: the one implemented in Moses, a phrase-based reordering similar to (Tillman, 2004), and our hierarchical reordering model (Galley and Manning, 2008). The models are similar in that they predict one of the following three orientations for each phrase: monotone, swap, and discontinuous. The hierarchical model improves our results by 0.25 BLEU, though the difference is not statistically significant.[4] We got an additional improvement of 0.28 BLEU by distinguishing between two types of discontinuities (i.e., left and right discontinuous), which yields 44.50 as our final BLEU score on MT08.

The critical value of the Google LM is shown in Table 2. Finally, Table 3 shows results for our primary and contrastive submissions. While the system incorporating the discriminative reordering model of (Zens and Ney, 2006) performed slightly worse on MT08, it was slightly more effective on MT09. Results on MT09 are those reported by NIST.

Acknowledgements

We thank our collaborators at IBM Research for preprocessing the parallel data used in this work. This paper is based on work funded in part by the Defense Advanced Research Projects Agency through IBM. The content does not necessarily reflect the views of the U.S. Government, and no official endorsement should be inferred.

Submissions	MT08	MT09
primary	44.50	47.81
contrastive	44.40	47.86

Table 3: BLEU-4[%] scores for primary and contrastive submissions.

References

Lalit R. Bahl, Frederick Jelinek, and Robert L. Mercer. 1983. A maximum likelihood approach to continuous speech recognition. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 5(2):179–190.

Peter F. Brown, Vincent J. Della Pietra, Stephen A. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: parameter estimation. *Comput. Linguist.*, 19(2):263–311.

Daniel Cer, Daniel Jurafsky, and Christopher D. Manning. 2008. Regularization and search for minimum error rate training. In *Proc. of the Third Workshop on Statistical Machine Translation*.

Stanley F. Chen and Joshua Goodman. 1996. An empirical study of smoothing techniques for language modeling. In *Proceedings of the Thirty-Fourth Annual Meeting of the Association for Computational Linguistics*, pages 310–318.

A. P. Dempster, N. M. Laird, and D. B. Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38.

Jenny Rose Finkel, Trond Grenager, and Christopher D. Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *ACL*.

Michel Galley and Christopher D. Manning. 2008. A simple and effective hierarchical phrase reordering model. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 848–856, Honolulu, Hawaii, October.

Michel Galley, Pi-Chuan Chang, Daniel Cer, Jenny Finkel, and Christopher Manning. 2008. Stanford University's Chinese-to-English statistical machine translation system for the 2008 NIST evaluation. In *Proceedings of the 2008 NIST Open Machine Translation Evaluation Workshop*, Arlington, Virginia, March.

[4] As shown in (Galley and Manning, 2008), which empirically evaluates all three models, gains provided by our hierarchical reordering model are typically more substantial on Chinese-English than on Arabic-English. This is understandable: the hierarchical model is aimed at improving *non-local* reordering, though most reorderings in Arabic-English tend to be local (especially with a distortion limit of 5).

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *NAACL '03: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*. Association for Computational Linguistics.

Philipp Koehn, Hieu Hoang, Alexandra Birch Mayne, Christopher Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Annual Meeting of the Association for Computation Linguistics (ACL), Demonstration Session*.

Shankar Kumar and William Byrne. 2004. Minimum bayes-risk decoding for statistical machine translation. In *Proc. of HLT-NAACL*.

P. Liang, B. Taskar, and D. Klein. 2006. Alignment by agreement. In *North American Association for Computational Linguistics (NAACL)*, pages 104–111.

Lucian Vlad Lita, Abe Ittycheriah, Salim Roukos, and Nanda Kambhatla. 2003. tRuEcasIng. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 152–159.

Einat Minkov, Richard Wang, Anthony Tomasic, and William Cohen. 2006. Ner systems that suit user's preferences: Adjusting the recall-precision trade-off for entity extraction. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, pages 93–96.

Franz Josef Och and Hermann Ney. 2004. The alignment template approach to statistical machine translation. *Computational Linguistics*, 30(4):417–449.

Franz Josef Och. 2002. *Statistical Machine Translation: From Single-Word Models to Alignment Templates*. Ph.D. thesis, RWTH Aachen.

Franz Josef Och. 2003. Minimum error rate training for statistical machine translation. In *ACL 2003: Proc. of the 41st Annual Meeting of the Association for Computational Linguistics*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2001. BLEU: a method for automatic evaluation of machine translation. In *ACL*.

Andreas Stolcke. 2002. SRILM – an extensible language modeling toolkit. In *Proc. Intl. Conf. on Spoken Language Processing (ICSLP–2002)*.

Christoph Tillman. 2004. A unigram orientation model for statistical machine translation. In *Proceedings of HLT-NAACL 2004: Short Papers*, pages 101–104.

Stephan Vogel, Hermann Ney, and Christoph Tillmann. 1996. HMM-based word alignment in statistical translation. In *Proceedings of the 16th conference on Computational linguistics*, pages 836–841.

Richard Zens and Herman Ney. 2006. Discriminative reordering models for statistical machine translation. In *Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL): Proceedings of the Workshop on Statistical Machine Translation*, pages 55–63, New York City, NY, June.