Feature-Rich Phrase-based Translation: Stanford University's Submission to the WMT 2013 Translation Task

Spence Green, Daniel Cer, Kevin Reschke, Rob Voigt[*]**, John Bauer**
Sida Wang, Natalia Silveira[†]**, Julia Neidert** and **Christopher D. Manning**
Computer Science Department, Stanford University
[*]Center for East Asian Studies, Stanford University
[†]Department of Linguistics, Stanford University
{spenceg,cerd,kreschke,robvoigt,horatio,sidaw,natalias,jneid,manning}@stanford.edu

Abstract

We describe the Stanford University NLP Group submission to the 2013 Workshop on Statistical Machine Translation Shared Task. We demonstrate the effectiveness of a new adaptive, online tuning algorithm that scales to large feature and tuning sets. For both English-French and English-German, the algorithm produces feature-rich models that improve over a dense baseline and compare favorably to models tuned with established methods.

1 Introduction

Green et al. (2013b) describe an online, adaptive tuning algorithm for feature-rich translation models. They showed considerable translation quality improvements over MERT (Och, 2003) and PRO (Hopkins and May, 2011) for two languages in a research setting. The purpose of our submission to the 2013 Workshop on Statistical Machine Translation (WMT) Shared Task is to compare the algorithm to more established methods in an evaluation. We submitted English-French (En-Fr) and English-German (En-De) systems, each with over 100k features tuned on 10k sentences. This paper describes the systems and also includes new feature sets and practical extensions to the original algorithm.

2 Translation Model

Our machine translation (MT) system is Phrasal (Cer et al., 2010), a phrase-based system based on alignment templates (Och and Ney, 2004). Like many MT systems, Phrasal models the predictive translation distribution $p(e|f;w)$ directly as

$$p(e|f;w) = \frac{1}{Z(f)} \exp\left[w^\top \phi(e,f)\right] \quad (1)$$

where e is the target sequence, f is the source sequence, w is the vector of model parameters, $\phi(\cdot)$ is a feature map, and $Z(f)$ is an appropriate normalizing constant. For many years the dimension of the feature map $\phi(\cdot)$ has been limited by MERT, which does not scale past tens of features.

Our submission explores real-world translation quality for high-dimensional feature maps and associated weight vectors. That case requires a more scalable tuning algorithm.

2.1 Online, Adaptive Tuning Algorithm

Following Hopkins and May (2011) we cast MT tuning as pairwise ranking. Consider a single source sentence f with associated references $e^{1:k}$. Let d be a derivation in an n-best list of f that has the target $e = e(d)$ and the feature map $\phi(d)$. Define the linear model score $M(d) = w \cdot \phi(d)$. For any derivation d_+ that is better than d_- under a gold metric G, we desire pairwise agreement such that

$$G\left(e(d_+), e^{1:k}\right) > G\left(e(d_-), e^{1:k}\right) \iff M(d_+) > M(d_-)$$

Ensuring pairwise agreement is the same as ensuring $w \cdot [\phi(d_+) - \phi(d_-)] > 0$.

For learning, we need to select derivation pairs (d_+, d_-) to compute difference vectors $x_+ = \phi(d_+) - \phi(d_-)$. Then we have a 1-class separation problem trying to ensure $w \cdot x_+ > 0$. The derivation pairs are sampled with the algorithm of Hopkins and May (2011). Suppose that we sample s pairs for source sentence f_t to compute a set of difference vectors $\mathcal{D}_t = \{x_+^{1:s}\}$. Then we optimize

$$\ell_t(w) = \ell(\mathcal{D}_t, w) = -\sum_{x_+ \in \mathcal{D}_t} \log \frac{1}{1 + e^{-w \cdot x_+}} \quad (2)$$

which is the familiar logistic loss. Hopkins and May (2011) optimize (2) in a batch algorithm that alternates between candidate generation (i.e., n-best list or lattice decoding) and optimization (e.g., L-BFGS). We instead use AdaGrad (Duchi et al., 2011), a variant of stochastic gradient descent (SGD) in which the learning rate is adapted to the data. Informally, AdaGrad scales the weight updates according to the geometry of the data observed in earlier iterations. Consider a particular dimension j of w, and let scalars $v_t = w_{t,j}$, $g_t = \nabla_j \ell_t(w_{t-1})$, and $G_t = \sum_{i=1}^{t} g_i^2$. The AdaGrad update rule is

$$v_t = v_{t-1} - \eta\, G_t^{-1/2} g_t \tag{3}$$

$$G_t = G_{t-1} + g_t^2 \tag{4}$$

In practice, G_t is a diagonal approximation. If $G_t = I$, observe that (3) is vanilla SGD.

In MT systems, the feature map may generate exponentially many irrelevant features, so we need to regularize (3). The L_1 norm of the weight vector is known to be an effective regularizer in such a setting (Ng, 2004). An efficient way to apply L_1 regularization is the Forward-Backward splitting (FOBOS) framework (Duchi and Singer, 2009), which has the following two-step update:

$$w_{t-\frac{1}{2}} = w_{t-1} - \eta_{t-1} \nabla \ell_{t-1}(w_{t-1}) \tag{5}$$

$$w_t = \arg\min_{w} \frac{1}{2} \|w - w_{t-\frac{1}{2}}\|_2^2 + \eta_{t-1} r(w) \tag{6}$$

where (5) is just an unregularized gradient descent step and (6) balances the regularization term $r(w)$ with staying close to the gradient step.

For L_1 regularization we have $r(w) = \lambda ||w||_1$ and the closed-form solution to (6) is

$$w_t = \text{sign}(w_{t-\frac{1}{2}}) \Big[|w_{t-\frac{1}{2}}| - \eta_{t-1}\lambda \Big]_+ \tag{7}$$

where $[x]_+ = \max(x, 0)$ is the clipping function that in this case sets a weight to 0 when it falls below the threshold $\eta_{t-1}\lambda$.

Online algorithms are inherently sequential; this algorithm is no exception. If we want to scale the algorithm to large tuning sets, then we need to parallelize the weight updates. Green et al. (2013b) describe the parallelization technique that is implemented in Phrasal.

2.2 Extensions to (Green et al., 2013b)

Sentence-Level Metric We previously used the gold metric BLEU+1 (Lin and Och, 2004), which smoothes bigram precisions and above. This metric worked well with multiple references, but we found that it is less effective in a single-reference setting like WMT. To make the metric more robust, Nakov et al. (2012) extended BLEU+1 by smoothing both the unigram precision and the reference length. We found that this extension yielded a consistent +0.2 BLEU improvement at test time for both languages. Subsequent experiments on the data sets of Green et al. (2013b) showed that standard BLEU+1 works best for multiple references.

Custom regularization parameters Green et al. (2013b) showed that large feature-rich models overfit the tuning sets. We discovered that certain features caused greater overfitting than others. Custom regularization strengths for each feature set are one solution to this problem. We found that technique largely fixed the overfitting problem as shown by the learning curves presented in section 5.1.

Convergence criteria Standard MERT implementations approximate tuning BLEU by reranking the previous n-best lists with the updated weight vector. This approximation becomes infeasible for large tuning sets, and is less accurate for algorithms like ours that do not accumulate n-best lists. We approximate tuning BLEU by maintaining the 1-best hypothesis for each tuning segment. At the end of each epoch, we compute corpus-level BLEU from this hypothesis set. We flush the set of stored hypotheses before the next epoch begins. Although memory-efficient, we find that this approximation is less dependable as a convergence criterion than the conventional method. Whereas we previously stopped the algorithm after four iterations, we now select the model according to held-out accuracy.

3 Feature Sets

3.1 Dense Features

The baseline "dense" model has 19 features: the nine Moses (Koehn et al., 2007) baseline features, a hierarchical lexicalized re-ordering model (Galley and Manning, 2008), the (log) bitext count of each translation rule, and an indicator for unique rules.

The final dense feature sets for each language differ slightly. The En-Fr system incorporates a second language model. The En-De system adds a future cost component to the linear distortion model (Green et al., 2010).The future cost estimate allows the distortion limit to be raised without a decrease in translation quality.

3.2 Sparse Features

Sparse features do not necessarily fire on each hypothesis extension. Unlike prior work on sparse MT features, our feature extractors do not filter features based on tuning set counts. We instead rely on the regularizer to select informative features.

Several of the feature extractors depend on source-side part of speech (POS) sequences and dependency parses. We created those annotations with the Stanford CoreNLP pipeline.

Discriminative Phrase Table A lexicalized indicator feature for each rule in a derivation. The feature weights can be interpreted as adjustments to the associated dense phrase table features.

Discriminative Alignments A lexicalized indicator feature for the phrase-internal alignments in each rule in a derivation. For one-to-many, many-to-one, and many-to-many alignments we extract the clique of aligned tokens, perform a lexical sort, and concatenate the tokens to form the feature string.

Discriminative Re-ordering A lexicalized indicator feature for each rule in a derivation that appears in the following orientations: monotone-with-next, monotone-with-previous, non-monotone-with-next, non-monotone-with-previous. Green et al. (2013b) included the richer non-monotone classes swap and discontinuous. However, we found that these classes yielded no significant improvement over the simpler non-monotone classes. The feature weights can be interpreted as adjustments to the generative lexicalized re-ordering model.

Source Content-Word Deletion Count-based features for source content words that are “deleted” in the target. Content words are nouns, adjectives, verbs, and adverbs. A deleted source word is either unaligned or aligned to one of the 100 most frequent target words in the target bitext. For each deleted word we increment both the feature for the particular source POS and an aggregate feature for all parts of speech. We add similar but separate features for head content words that are either unaligned or aligned to frequent target words.

Inverse Document Frequency Numeric features that compare source and target word frequencies. Let idf$(\cdot)$ return the inverse document frequency of a token in the training bitext. Suppose a derivation $d = \{r_1, r_2, \ldots, r_n\}$ is composed of n translation rules, where $e(r)$ is the target side of the rule and $f(r)$ is the source side. For each rule r that translates j source tokens to i target tokens we compute

$$q = \sum_i \text{idf}(e(r)_i) - \sum_j \text{idf}(f(r)_j) \qquad (8)$$

We add two numeric features, one for the source and another for the target. When $q > 0$ we increment the target feature by q; when $q < 0$ we increment the target feature by $|q|$. Together these features penalize asymmetric rules that map rare words to frequent words and vice versa.

	Bilingual		Monolingual
	Sentences	*Tokens*	*Tokens*
En-Fr	5.0M	289M	1.51B
En-De	4.4M	223M	1.03B

Table 1: Gross corpus statistics after data selection and pre-processing. The En-Fr monolingual counts include French Gigaword 3 (LDC2011T10).

POS-based Re-ordering The lexicalized discriminative re-ordering model is very sparse, so we added re-ordering features based on source parts of speech. When a rule is applied in a derivation, we extract the associated source POS sequence along with the POS sequences from the previous and next rules. We add a “with-previous” indicator feature that is the conjunction of the current and previous POS sequences; the “with-next” indicator feature is created analogously. This feature worked well for En-Fr, but not for En-De.

4 Data Preparation

Table 1 describes the pre-processed corpora from which our systems are built.

4.1 Data Selection

We used all of the monolingual and parallel En-De data allowed in the constrained condition. We incorporated all of the French monolingual data, but sampled a 5M-sentence bitext from the approximately 40M available En-Fr parallel sentences. To select the sentences we first created a “target” corpus by concatenating the tuning and test sets (newstest2008–2013). Then we ran the feature decay algorithm (FDA) (Biçici and Yuret, 2011), which samples sentences that most closely resemble the target corpus. FDA is a principled method for reducing the phrase table size by excluding less relevant training examples.

4.2 Tokenization

We tokenized the English (source) data according to the Penn Treebank standard (Marcus et al., 1993) with Stanford CoreNLP. The French data was tokenized with packages from the Stanford French Parser (Green et al., 2013a), which implements a scheme similar to that used in the French Treebank (Abeillé et al., 2003).

German is more complicated due to pervasive compounding. We first tokenized the data with the same English tokenizer. Then we split compounds with the lattice-based model (Dyer, 2009) in cdec (Dyer et al., 2010). To simplify post-processing we added segmentation markers to split tokens, e.g., *überschritt ⇒ über #schritt*.

4.3 Alignment

We aligned both bitexts with the Berkeley Aligner (Liang et al., 2006) configured with standard settings. We symmetrized the alignments according to the grow-diag heuristic.

4.4 Language Modeling

We estimated unfiltered 5-gram language models using lmplz (Heafield et al., 2013) and loaded them with KenLM (Heafield, 2011). For memory efficiency and faster loading we also used KenLM to convert the LMs to a trie-based, binary format. The German LM included all of the monolingual data plus the target side of the En-De bitext. We built an analogous model for French. In addition, we estimated a separate French LM from the Gigaword data.[1]

4.5 French Agreement Correction

In French verbs must agree in number and person with their subjects, and adjectives (and some past participles) must agree in number and gender with the nouns they modify. On their own, phrasal alignment and target side language modeling yield correct agreement inflection most of the time. For verbs, we find that the inflections are often accurate: number is encoded in the English verb and subject, and 3rd person is generally correct in the absence of a 1st or 2nd person pronoun. However, since English does not generally encode gender, adjective inflection must rely on language modeling, which is often insufficient.

To address this problem we apply an automatic inflection correction post-processing step. First, we generate dependency parses of our system's output using BONSAI (Candito and Crabbé, 2009), a French-specific extension to the Berkeley Parser (Petrov et al., 2006). Based on these dependencies, we match adjectives with the nouns they modify and past participles with their subjects. Then we use *Lefff* (Sagot, 2010), a machine-readable French lexicon, to determine the gender and number of the noun and to choose the correct inflection for the adjective or participle.

Applied to our 3,000 sentence development set, this correction scheme produced 200 corrections with perfect accuracy. It produces a slight (-0.014) drop in BLEU score. This arises from cases where the reference translation uses a synonymous but differently gendered noun, and consequently has different adjective inflection.

4.6 German De-compounding

Split German compounds must be merged after translation. This process often requires inserting affixes (e.g., *s*, *en*) between adjacent tokens in the compound. Since the German compounding rules are complex and exception-laden, we rely on a dictionary lookup procedure with backoffs. The dictionary was constructed during pre-processing. To compound the final translations, we first lookup the compound sequence—which is indicated by segmentation markers—in the dictionary. If it is present, then we use the dictionary entry. If the compound is novel, then for each pair of words to be compounded, we insert the suffix most commonly appended in compounds to the first word of the pair. If the first word itself is unknown in our dictionary, we insert the suffix most commonly appended after the last three characters. For example, words ending with *ung* most commonly have an *s* appended when they are used in compounds.

4.7 Recasing

Phrasal includes an LM-based recaser (Lita et al., 2003), which we trained on the target side of the bitext for each language. On the newstest2012 development data, the German recaser was 96.8% accurate and the French recaser was 97.9% accurate.

5 Translation Quality Experiments

During system development we tuned on newstest2008–2011 (10,570 sentences) and tested

[1] The MT system learns significantly different weights for the two LMs: 0.086 for the primary LM and 0.044 for the Gigaword LM.

	#iterations	#features	tune	newstest2012	newstest2013†
Dense	10	20	30.26	31.12	–
Feature-rich	11	207k	32.29	31.51	29.00

Table 2: En-Fr BLEU-4 [% uncased] results. The tuning set is newstest2008–2011. (†) newstest2013 is the cased score computed by the WMT organizers.

	#iterations	#features	tune	newstest2012	newstest2013†
Dense	10	19	16.83	18.45	–
Feature-rich	13	167k	17.66	18.70	18.50

Table 3: En-De BLEU-4 [% uncased] results.

on newstest2012 (3,003 sentences). We compare the feature-rich model to the "dense" baseline.

The En-De system parameters were: 200-best lists, a maximum phrase length of 8, and a distortion limit of 6 with future cost estimation. The En-Fr system parameters were: 200-best lists, a maximum phrase length of 8, and a distortion limit of 5.

The online tuning algorithm used a default learning rate $\eta = 0.03$ and a mini-batch size of 20. We set the regularization strength λ to 10.0 for the discriminative re-ordering model, 0.0 for the dense features, and 0.1 otherwise.

5.1 Results

Tables 2 and 3 show En-Fr and En-De results, respectively. The "Feature-rich" model, which contains the full complement of dense and sparse features, offers a meager improvement over the "Dense" baseline. This result contrasts with the results of Green et al. (2013b), who showed significant translation quality improvements over the same dense baseline for Arabic-English and Chinese-English. However, they had multiple target references, whereas the WMT data sets have just one. We speculate that this difference is significant. For example, consider a translation rule that rewrites to a 4-gram in the reference. This event can increase the sentence-level score, thus encouraging the model to upweight the rule indicator feature.

More evidence of overfitting can be seen in Figure 1, which shows learning curves on the development set for both language pairs. Whereas the dense model converges after just a few iterations, the feature-rich model continues to creep higher. Separate experiments on a held-out set showed that generalization did not improve after about eight iterations.

6 Conclusion

We submitted a feature-rich MT system to WMT 2013. While sparse features did offer a measurable improvement over a baseline dense feature set, the gains were not as significant as those shown by Green et al. (2013b). One important difference between the two sets of results is the number of references. Their NIST tuning and test sets had four references; the WMT data sets have just one. We speculate that sparse features tend to overfit more in this setting. Individual features can greatly influence the sentence-level metric and thus become large components of the gradient. To combat this phenomenon we experimented with custom regularization strengths and a more robust sentence-level metric. While these two improvements greatly reduced the model size relative to (Green et al., 2013b), a generalization problem remained. Nevertheless, we showed that feature-rich models are now competitive with the state-of-the-art.

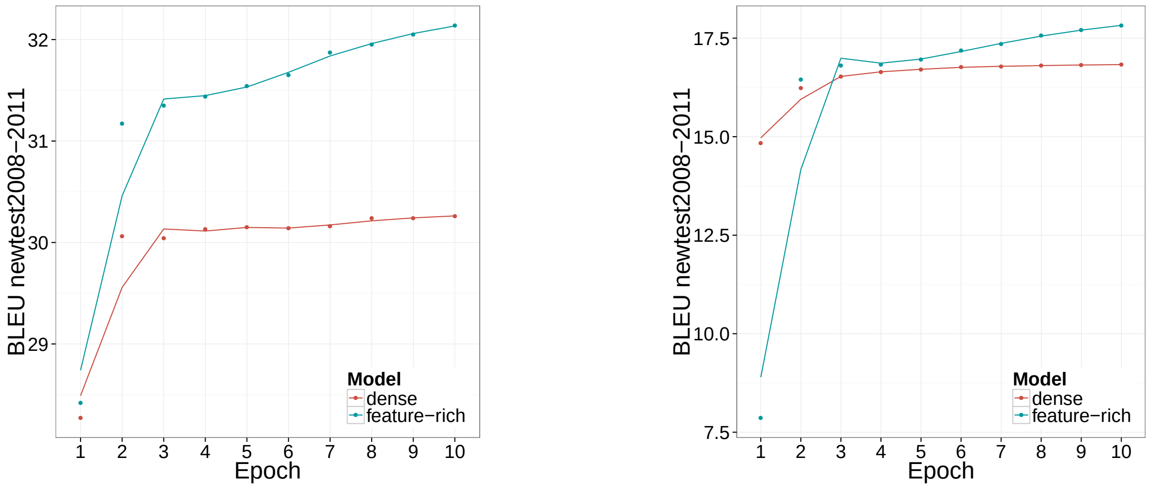
Acknowledgments This work was supported by the Defense Advanced Research Projects Agency (DARPA) Broad Operational Language Translation (BOLT) program through IBM. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the view of DARPA or the US government.

References

A. Abeillé, L. Clément, and A. Kinyon, 2003. *Building a treebank for French*, chapter 10. Kluwer.

E. Biçici and D. Yuret. 2011. Instance selection for machine translation using feature decay algorithms. In *WMT*.

M. Candito and B. Crabbé. 2009. Improving generative statistical parsing with semi-supervised word clustering. In *IWPT*.



(a) En-Fr tuning

(b) En-De tuning

Figure 1: BLEU-4 [% uncased] Learning curves on newstest2008–2011 with loess trend lines.

D. Cer, M. Galley, D. Jurafsky, and C. D. Manning. 2010. Phrasal: A statistical machine translation toolkit for exploring new model features. In *HLT-NAACL, Demonstration Session*.

J. Duchi and Y. Singer. 2009. Efficient online and batch learning using forward backward splitting. *JMLR*, 10:2899–2934.

J. Duchi, E. Hazan, and Y. Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *JMLR*, 12:2121–2159.

C. Dyer, A. Lopez, J. Ganitkevitch, J. Weese, F. Ture, et al. 2010. `cdec`: A decoder, alignment, and learning framework for finite-state and context-free translation models. In *ACL System Demonstrations*.

C. Dyer. 2009. Using a maximum entropy model to build segmentation lattices for MT. In *NAACL*.

M. Galley and C. D. Manning. 2008. A simple and effective hierarchical phrase reordering model. In *EMNLP*.

S. Green, M. Galley, and C. D. Manning. 2010. Improved models of distortion cost for statistical machine translation. In *HLT-NAACL*.

S. Green, M-C. de Marneffe, and C. D. Manning. 2013a. Parsing models for identifying multiword expressions. *Computational Linguistics*, 39(1):195–227.

S. Green, S. Wang, D. Cer, and C. D. Manning. 2013b. Fast and adaptive online training of feature-rich translation models. In *ACL*.

K. Heafield, I. Pouzyrevsky, J. H. Clark, and P. Koehn. 2013. Scalable modified Kneser-Ney language model estimation. In *ACL, Short Papers*.

K. Heafield. 2011. KenLM: Faster and smaller language model queries. In *WMT*.

M. Hopkins and J. May. 2011. Tuning as ranking. In *EMNLP*.

P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *ACL, Demonstration Session*.

P. Liang, B. Taskar, and D. Klein. 2006. Alignment by agreement. In *NAACL*.

C.-Y. Lin and F. J. Och. 2004. ORANGE: a method for evaluating automatic evaluation metrics for machine translation. In *COLING*.

L. V. Lita, A. Ittycheriah, S. Roukos, and N. Kambhatla. 2003. tRuEcasIng. In *ACL*.

M. Marcus, M. A. Marcinkiewicz, and B. Santorini. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19:313–330.

P. Nakov, F. Guzman, and S. Vogel. 2012. Optimizing for sentence-level BLEU+1 yields short translations. In *COLING*.

A. Y. Ng. 2004. Feature selection, L_1 vs. L_2 regularization, and rotational invariance. In *ICML*.

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

F. J. Och. 2003. Minimum error rate training for statistical machine translation. In *ACL*.

S. Petrov, L. Barrett, R. Thibaux, and D. Klein. 2006. Learning accurate, compact, and interpretable tree annotation. In *ACL*.

B. Sagot. 2010. The Lefff, a freely available and large-coverage morphological and syntactic lexicon for French. In *LREC*.