An Empirical Comparison of Features and Tuning for Phrase-based Machine Translation

Spence Green, Daniel Cer, and **Christopher D. Manning**
Computer Science Department, Stanford University
{spenceg,danielcer,manning}@stanford.edu

Abstract

Scalable discriminative training methods are now broadly available for estimating phrase-based, feature-rich translation models. However, the sparse feature sets typically appearing in research evaluations are less attractive than standard dense features such as language and translation model probabilities: they often overfit, do not generalize, or require complex and slow feature extractors. This paper introduces *extended features*, which are more specific than dense features yet more general than lexicalized sparse features. Large-scale experiments show that extended features yield robust BLEU gains for both Arabic-English (+1.05) and Chinese-English (+0.67) relative to a strong feature-rich baseline. We also specialize the feature set to specific data domains, identify an objective function that is less prone to overfitting, and release fast, scalable, and language-independent tools for implementing the features.

1 Introduction

Scalable discriminative algorithm design for machine translation (MT) has lately been a booming enterprise. There are now algorithms for every taste: probabilistic and distribution-free, online and batch, regularized and unregularized. Technical differences aside, the papers that apply these algorithms to phrase-based translation often share a curious empirical characteristic: the algorithms support extra features, but the features do not significantly improve translation. For example, Hopkins and May (2011) showed that PRO with some simple ad hoc features only exceeds the baseline on one of three language pairs. Gimpel and Smith (2012b) observed a similar result for both PRO and their ramp-loss algorithm. Cherry and Foster (2012) found that, at least in the batch case, many algorithms produce similar results, and features only significantly increased quality for one of three language pairs. Only recently did Cherry (2013) and Green et al. (2013b) identify certain features that consistently reduce error.

These empirical results suggest that feature design and model fitting, the subjects of this paper, warrant a closer look. We introduce an effective *extended feature* set for phrase-based MT and identify a loss function that is less prone to overfitting. Extended features share three attractive characteristics with the standard Moses *dense features* (Koehn et al., 2007): ease of implementation, language independence, and independence from ancillary corpora like treebanks. In our experiments, they do not overfit and can be extracted efficiently during decoding. Because all feature weights are tuned on the development set, the new feature templates are amenable to feature augmentation (Daumé III, 2007), a simple domain adaptation technique that we show works surprisingly well for MT.

Extended features are designed according to a principle rather than a rule: they should fire less than standard dense features, which are general, but more than so-called *sparse features*, which are very specific—they are usually lexicalized—and thus prone to overfitting. This principle is motivated by analysis, which shows how expressive models can be a mixed blessing in the translation setting. It is obvious that features allow the model to fit the tuning data more tightly. For example, sparse lexicalized features could reduce tuning error by learning that the references prefer *U.S.* over *United States*, a minor lexical distinction. Reference choice should matter more than in the dense case, an issue that we quantify. We also show that frequency cutoffs, which are a crude but common form of feature selection, are unnecessary and even detrimental when features follow this principle.

We report large-scale translation quality experiments relative to both dense and feature-rich baselines. Our best feature set, which includes domain adaptation features, yields an average +1.05 BLEU improvement for Arabic-English and +0.67 for

Proceedings of the Ninth Workshop on Statistical Machine Translation, pages 466–476,
Baltimore, Maryland USA, June 26–27, 2014.

Chinese-English. In addition to the extended feature set, we show that an online variant of expected error (Och, 2003) is significantly faster to compute, less prone to overfitting, and nearly as effective as a pairwise loss. We release all software—feature extractors, and fast word clustering and data selection packages—used in our experiments.[1]

2 Phrase-based Models and Learning

The log-linear approach to phrase-based translation (Och and Ney, 2004) directly models the predictive translation distribution

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

where e is the target string, f is the source string, $w \in \mathbb{R}^d$ is the vector of model parameters, $\phi(\cdot) \in \mathbb{R}^d$ is a feature map, and $Z(f)$ is an appropriate normalizing constant. Assume that there is also a function $\rho(e,f) \in \mathbb{R}^d$ that produces a recombination map for the features. That is, each coordinate in ρ represents the state of the corresponding coordinate in ϕ. For example, suppose that ϕ_j is the log probability produced by the n-gram language model (LM). Then ρ_j would be the appropriate LM history. Recall that recombination collapses derivations with equivalent recombination maps during search and thus affects learning. This issue significantly influences feature design.

To learn w, we follow the online procedure of Green et al. (2013b), who calculate gradient steps with AdaGrad (Duchi et al., 2011) and perform feature selection via L_1 regularization in the FOBOS (Duchi and Singer, 2009) framework. This procedure accommodates any loss function for which a subgradient can be computed. Green et al. (2013b) used a PRO objective (Hopkins and May, 2011) with a logistic (surrogate) loss function. However, later results showed overfitting (Green et al., 2013a), and we found that their online variant of PRO tends to produce short translations like its batch counterpart (Nakov et al., 2013). Moreover, PRO requires sampling, making it slow to compute.

To address these shortcomings, we explore an online variant of expected error (Och, 2003, Eq.7). Let $\mathbf{E}_t = \{e_i\}_{i=1}^n$ be a scored n-best list of translations at time step t for source input f_t. Let $G(e)$ be a gold error metric that evaluates each candidate translation with respect to a set of one or more references. The smooth loss function is

$$\begin{aligned}\ell_t(w_{t-1}) &= E_{p(e|f_t;w_{t-1})}[G(e)] \\ &= \frac{1}{Z} \sum_{e' \in \mathbf{E}_t} \exp\left(w^\top \phi(e',f)\right) \cdot G(e')\end{aligned} \quad (2)$$

with normalization constant $Z = \sum_{e' \in \mathbf{E}_t} \exp\left(w^\top \phi(e',f)\right)$. The gradient g_t for coordinate j is:

$$\begin{aligned}g_t = E[G(e)\phi_j(e,f_t)] - \\ E[G(e)]E[\phi_j(e,f_t)]\end{aligned} \quad (3)$$

To our knowledge, we are the first to experiment with the online version of this loss.[2] When $G(e)$ is sentence-level BLEU+1 (Lin and Och, 2004)—the setting in our experiments—this loss is also known as expected BLEU (Cherry and Foster, 2012). However, other metrics are possible.

3 Extended Phrase-based Features

We divide our feature templates into five categories, which are well-known sources of error in phrase-based translation. The features are defined over derivations $d = \{r_i\}_{i=1}^D$, which are ordered sequences of rules r from the translation model. Define functions $f(\cdot)$ to be the source string of a rule or derivation and $e(\cdot)$ to be the target string. *Local features* can be extracted from individual rules and do not declare any state in the recombination map, thus for all local features i we have $\rho_i = 0$. *Nonlocal features* are defined over partial derivations and declare some state, either a real-valued parameter or an index indicating a categorical value like an n-gram context.

For each language, the extended feature templates require unigram counts and a word-to-class mapping $\varphi : w \mapsto c$ for word $w \in V$ and class $c \in C$. These can be extracted from any monolingual data; our experiments simply use both sides of the unaligned parallel training data.

The features are language-independent, but we will use Arabic-English as a running example.

3.1 Lexical Choice

Lexical choice features make more specific distinctions between target words than the dense translation model features (Koehn et al., 2003).

[1] http://nlp.stanford.edu/software/phrasal

[2] Gao and He (2013) used stochastic gradient descent and expected BLEU to learn phrase table feature weights, but not the full translation model w.

Lexicalized rule indicator (Liang et al., 2006a) Some rules occur frequently enough that we can learn rule-specific weights that augment the dense translation model features. For example, our model learns the following rule indicator features and weights:

أسباب ⇒ reasons	-0.022
أسباب ⇒ reasons for	0.002
أسباب ⇒ the reasons for	0.016

These translations are all correct depending on context. When the plural noun أسباب 'reasons' appears in a construct state (*iDafa*) the preposition *for* is unrealized. Moreover, depending on the context, the English translation might also require the determiner *the*, which is also unrealized. The weights reflect that أسباب 'reasons' often appears in construct and boost insertion of necessary target terms. To prevent overfitting, this template only fires an indicator for rules that occur more than 50 times in the parallel training data (this is different from frequency filtering on the tuning data; see section 6.1). The feature is local.

Class-based rule indicator Word classes abstract over lexical items. For each rule r, a *prototype* that abstracts over many rules can be built by concatenating $\{\varphi(w) : w \in f(r)\}$ with $\{\varphi(w) : w \in e(r)\}$. For example, suppose that Arabic class 492 consists primarily of Arabic present tense verbs and class 59 contains English auxiliaries. Then the model might penalize a rule prototype like `492>59_59`, which drops the verb. This template fires an indicator for each rule prototype and is local.

Target unigram class (Ammar et al., 2013) Target lexical items with similar syntactic and semantic properties may have very different frequencies in the training data. These frequencies will influence the dense features. For example, in one of our English class mappings the following words map to the same class:

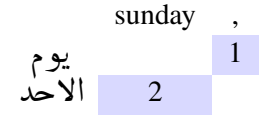
word	class	freq.
surface-to-surface	0	269
air-to-air	0	98
ground-to-air	0	63

The classes capture common linguistic attributes of these words, which is the motivation for a full class-based LM. Learning unigram weights directly is surprisingly effective and does not require building another LM. This template fires a separate indicator for each class $\{\varphi(w) : w \in e(r)\}$ and is local.

3.2 Word Alignments

Word alignment features allow the model to recognize fine-grained phrase-internal information that is largely opaque in the dense model.

Lexicalized alignments (Liang et al., 2006a) Consider the internal alignments of the rule:



Alignment 1 ⟨يوم 'day' ⇒ ,⟩ is incorrect and alignment 2 is correct. The dense translation model features might assign this rule high probability if alignment 1 is a common alignment error. Lexicalized alignment features allow the model to compensate for these events. This feature fires an indicator for each alignment in a rule—including multiword cliques—and is local.

Class-based alignments Like the class-based rule indicator, this feature template replaces each lexical item with its word class, resulting in an alignment prototype. This feature fires an indicator for each alignment in a rule after mapping lexical items to classes. It is local.

Source class deletion Phrase extraction algorithms often use a "grow" symmetrization step (Och and Ney, 2003) to add alignment points. Sometimes this procedure can produce a rule that deletes important source content words. This feature template allows the model to penalize these rules by firing an indicator for the class of each unaligned source word. The feature is local.

Punctuation ratio Languages use different types and ratios of punctuation (Salton, 1958). For example, quotation marks are not commonly used in Arabic, but they are conventional in English. Furthermore, spurious alignments often contain punctuation. To control these two phenomena, this feature template returns the ratio of target punctuation tokens to source punctuation tokens for each derivation. Since the denominator is constant, this feature can be computed incrementally as a derivation is constructed. It is local.

Function word ratio Words can also be spuriously aligned to non-punctuation, non-digit function words such as determiners and particles. Furthermore, linguistic differences may account for

differences in function word occurrences. For example, English has a broad array of modal verbs and auxiliaries not found in Arabic. This feature template takes the 25 most frequent words in each language (according to the unigram counts), and computes the ratio between target and source function words for each derivation. As before the denominator is constant, so the feature can be computed efficiently. It is local.

3.3 Phrase Boundaries

The LM and hierarchical reordering model are the only dense features that cross phrase boundaries.

Target-class bigram boundary We have already added target class unigrams. We find that both lexicalized and class-based bigrams cause overfitting, therefore we restrict to bigrams that straddle phrase boundaries. The feature template fires an indicator for the concatenation of the word classes on either side of each boundary. This feature is non-local and its recombination state ρ is the word class at the right edge of the partial derivation.

3.4 Derivation Quality

To satisfy strong features like the LM, or hard constraints like the distortion limit, the phrase-based model can build derivations from poor translation rules. For example, a derivation consisting mostly of unigram rules may miss idiomatic usage that larger rules can capture. All of these feature templates are local.

Source dimension (Hopkins and May, 2011) An indicator feature for the source dimension of the rule: $|f(r)|$.

Target dimension (Hopkins and May, 2011) An indicator for the target dimension: $|e(r)|$.

Rule shape (Hopkins and May, 2011) The conjunction of source and target dimension: $|f(r)|\_|e(r)|$.

3.5 Reordering

Lexicalized reordering models score the orientation of a rule in an alignment grid. We use the same baseline feature extractor as Moses, which has three classes: monotone, swap, and discontinuous. We also add the non-monotone class, which is a conjunction of swap and discontinuous, for a total of eight orientations.[3]

[3]Each class has "with-previous" and "with-next" specializations.

Algorithm (implementation)	#threads	Time
Brown (wcluster)	1	1023.39
Clark (cluster_neyessen)	1	890.11
Och (mkcls)	1	199.04
PredictiveFull (this paper)	8	3.27
Predictive (this paper)	8	2.42

Table 1: Wallclock time (min.sec) to generate a mapping from a vocabulary of 63k English words (3.7M tokens) to 512 classes. All experiments were run on the same server, which had eight physical cores. Our Java implementation is multi-threaded; the C++ baselines are single-threaded.

Lexicalized rule orientation (Liang et al., 2006a) For each rule, the template fires an indicator for the concatenation of the orientation class, each element in $f(r)$, and each element in $e(r)$. To prevent overfitting, this template only fires for rules that occur more than 50 times in the training data. The feature is non-local and its recombination state ρ is the rule orientation.

Class-based rule orientation For each rule, the template fires an indicator for the concatenation of the orientation class, each element in $\{\varphi(w) : w \in f(r)\}$, and each element in $\{\varphi(w) : w \in e(r)\}$. The feature is non-local and its recombination state ρ is the rule orientation.

Signed linear distortion The dense feature set includes a simple reordering cost model. Assume that $[r]$ returns the index of the leftmost source index in $f(d)$ and $[[r]]$ returns the rightmost index. Then the linear distortion is:

$$\delta = [r_1] + \sum_{i=2}^{D} |[[r_{i-1}]] + 1 - [r_i]| \qquad (4)$$

This score does not distinguish between left and right distortion. To correct this issue, this feature template fires an indicator for each signed component in the sum, for each positive and negative component. The feature is non-local and its recombination state ρ is the signed distortion.

3.6 Feature Dependencies

While unigram counts are trivial to compute, the same is not necessarily true of the word-to-class mapping φ. Standard algorithms run in $O(n^2)$, where $n = |V|$. Table 1 shows an evaluation of standard implementations of several popular algorithms: **Brown** et al. (1992) implemented by Liang

(2005); **Clark** (2003) without the morphological prior, which increases training time dramatically; and the implementation of **Och** (1999) that comes with the GIZA++ word aligner. The latter has been used recently for MT features (Ammar et al., 2013; Cherry, 2013; Yu et al., 2013). In a broad survey, Christodoulopoulos et al. (2010) found that for several downstream tasks, most word clustering algorithms—including Brown and Clark—result in similar task accuracy. For our large-scale setting, the primary issue is then the time to estimate φ.

For large corpora the existing implementations may require days or weeks, making our feature set less practical than the traditional dense MT features. Consequently, we re-implemented the predictive one-sided class model of Whittaker and Woodland (2001) with the parallelized clustering algorithm of Uszkoreit and Brants (2008) (**Predictive**), which was originally developed for very large scale language modeling. Our implementation uses multiple threads on a single processor instead of MapReduce. We also added two extensions that are useful for translation features. First, we map all digits to 0. This reduces sparsity while retaining useful patterns such as *0000* (e.g., years) and *0th* (e.g., ordinals). Second, we mapped all words occurring fewer than τ times to an `<unk>` token. In our experiment, these two changes reduce the vocabulary size by 71.1%. They also make the mapping φ more robust to unseen events during translation decoding. For a conservative comparison to the other three algorithms, we include results without these two extensions (**PredictiveFull**).[4]

4 Domain Adaptation Features

Feature augmentation is a simple yet effective domain adaptation technique (Daumé III, 2007). Suppose that the source data comes from M domains. Then for each original feature ϕ_i, we add M additional features, one for each domain. The original feature ϕ_i can be interpreted as a prior over the M domains (Finkel and Manning, 2009, fn.2).

Most of the extended features are defined over rules, so the critical issue is how to identify in-domain rules. The trick is to know which training sentence pairs are in-domain. Then we can annotate all rules extracted from these instances with domain labels. The in-domain rule sets need not be disjoint since some rules might be useful across domains.

This paper explores the following approach: we choose one of the M domains as the default. Next, we collect some source sentences for each of the $M-1$ remaining domains. Using these examples we then identify in-domain sentence pairs in the bitext via data selection, in our case the feature decay algorithm (Biçici and Yuret, 2011). Finally, our rule extractor adds domain labels to all rules extracted from each selected sentence pair. Crucially, these labels do not influence which rules are extracted or how they are scored. The resulting phrase table contains the same rules, but with a few additional annotations.

Our method assumes domain labels for each source input to be decoded. Our experiments utilize gold, document-level labels, but accurate sentence-level domain classifiers exist (Wang et al., 2012).

4.1 Augmentation of Extended Features

Irvine et al. (2013) showed that lexical selection is the most quantifiable and perhaps most common source of error in phrase-based domain adaptation. Our development experiments seemed to confirm this hypothesis as augmentation of the class-based and non-lexical (e.g., Rule shape) features did not reduce error. Therefore, we only augment the lexicalized features: rule indicators and orientations, and word alignments.

4.2 Domain-Specific Feature Templates

In-domain Rule Indicator (Durrani et al., 2013) An indicator for each rule that matches the input domain. This template fires a generic in-domain indicator and a domain-specific indicator (e.g., the features might be `indomain` and `indomain-nw`). The feature is local.

Adjacent Rule Indicator Indicators for adjacent in-domain rules. This template also fires both generic and domain-specific features. The feature is non-local and the state is a boolean indicating if the last rule in a partial derivation is in-domain.

5 Experiments

We evaluate and analyze our feature set under a variety of large-scale experimental conditions including multiple domains and references. To our knowledge, the only language pairs with sufficient research resources to support this protocol are Arabic-English (Ar-En) and Chinese-English (Zh-En). The

[4] For the baselines the training settings are the suggested defaults: Brown, default; Clark, 10 iterations, frequency cutoff $\tau = 5$; Och, 10 iterations. Our implementation: PredictiveFull, 30 iterations, $\tau = 0$; Predictive, 30 iterations, $\tau = 5$.

	Bilingual		Monolingual
	#Seg.	*#Tok.*	*#Tok.*
Ar-En	6.6M	375M	990M
Zh-En	9.3M	538M	

Table 2: Bilingual and monolingual training corpora. The monolingual English data comes from the AFP and Xinhua sections of English Gigaword 4 (LDC2009T13).

	#Seg.		#Ref.	Domains
	Ar-En	Zh-En		
tune	5,604	5,900	4	nw,wb,bn
dev	1,075	1,597	4	nw
dev-dom	2,203	2,317	1	nw,wb,bn
test1	1,313	820	4	nw,wb
test2	1,378	1,370	4	nw,wb
test3	628	613	1	nw,wb,bn

Table 3: Development, test, and tuning data. Domain abbreviations: broadcast news (**bn**), newswire (**nw**), and web (**wb**).

training corpora[5] come from several Linguistic Data Consortium (LDC) sources from 2012 and earlier (Table 2). The test, development, and tuning corpora[6] come from the NIST OpenMT and MetricsMATR evaluations (Table 3). Extended features benefit from more tuning data, so we concatenated five NIST data sets to build one large tuning set. Observe that all test data come from later epochs than the tuning and development data.

From these data we built phrase-based MT systems with Phrasal (Green et al., 2014).[7] We aligned the parallel corpora with the Berkeley aligner (Liang et al., 2006b) with standard settings and symmetrized via the grow-diag heuristic. We created separate English LMs for each language pair by concatenating the monolingual Gigaword data with the target-side of the respective bitexts. For each corpus we estimated unfiltered 5-gram language models with lmplz (Heafield et al., 2013).

For each condition we ran the learning algorithm for 25 epochs[8] and selected the model according to the maximum uncased, corpus-level BLEU-4 (Papineni et al., 2002) score on the dev set.

5.1 Results

We evaluate the new feature set relative to two baselines. **Dense** is the same baseline as Green et al. (2013b); these dense features are included in all of the models that follow. **Sparse** is their best feature-rich model, which adds lexicalized rule indicators, alignments, orientations, and source deletions without bitext frequency filtering.

We do not perform a full ablation study. Both the approximate search and the randomization of the order of tuning instances make the contributions of each individual template differ from run to run. Resource constraints prohibit multiple large-scale runs for each incremental feature. Instead, we divide the extended feature set into two parts, and report large-scale results. **Ext** includes all extended features except for the the filtered lexicalized feature templates. **Ext+Filt** adds those filtered lexicalized templates: rule indicators and orientations, and word alignments (section 3).

Table 4 shows translation quality results. The new feature set significantly exceeds the baseline Dense model for both language pairs. An interesting result is that the new extended features alone match the strong Sparse baseline. The class-based features, which are more general, should clearly be preferred to the sparse features when decoding out-of-domain data (so long as word mappings are trained for that data). The increased runtime per iteration comes not from feature extraction but from larger inner products as the model size increases.

Next, we add the domain features from section 4.2. We marked in-domain sentence pairs by concatenating the tuning data with additional bn and wb monolingual in-domain data from several LDC sources.[9] The FDA selection size was set to 20 times the number of in-domain examples for each genre. Newswire was selected as the default domain since most of the bitext comes from that domain.

The bottom rows of Tables 4a and 4b compare

[5]We tokenized the English with Stanford CoreNLP according to the Penn Treebank standard (Marcus et al., 1993), the Arabic with the Stanford Arabic segmenter (Monroe et al., 2014) according to the Penn Arabic Treebank standard (Maamouri et al., 2008), and the Chinese with the Stanford Chinese segmenter (Chang et al., 2008) according to the Penn Chinese Treebank standard (Xue et al., 2005).

[6]Data sources: tune, MT023568; dev, MT04; dev-dom, domain adaptation dev set is MT04 and all wb and bn data from LDC2007E61; test1, MT09 (Ar-En) and MT12 (Zh-En); test2, Progress0809 which was revealed in the OpenMT 2012 evaluation; test3, MetricsMATR08-10.

[7]System settings: distortion limit of 5, cube pruning beam size of 1200, maximum phrase length of 7.

[8]Other learning settings: 16 threads, mini-batch size of 20; L_1 regularization strength $\lambda = 0.001$; learning rate $\eta_0 = 0.02$; initialization of LM to 0.5, word penalty to -1.0, and all other dense features to 0.2; initialization of extended features to 0.0.

[9]Catalog: LDC2007T24, LDC2008T08, LDC2008T18, LDC2012T16, LDC2013T01, LDC2013T05, LDC2013T14.

Model	#features	Epochs	Min. / Epoch	tune	dev	test1	test2	test3
DENSE (D)	18	24	3	49.52	50.25	47.98	43.41	27.56
D+SPARSE	48,597	24	8	56.51	52.98	49.55	45.40	29.02
D+EXT	62,931	16	11	57.83	54.33	49.66	45.66	29.15
D+EXT+FILT	94,606	17	14	59.13	55.35	50.02	46.24	29.59
D+EXT+FILT+DOM	123,353	22	18	59.97	29.20†	**50.45**	**46.24**	**30.84**

(a) Ar-En.

Model	#features	Epochs	Min. / Epoch	tune	dev	test1	test2	test3
DENSE (D)	18	17	3	32.82	34.96	26.61	26.72	10.19
D+SPARSE	55,024	17	8	38.91	36.68	27.86	28.41	10.98
D+EXT	67,936	16	13	40.96	37.19	28.27	28.40	10.72
D+EXT+FILT	100,275	17	14	41.38	37.36	28.68	28.90	11.24
D+EXT+FILT+DOM	126,014	17	14	41.70	17.20†	***28.71***	***28.96***	***11.67***

(b) Zh-En.

Table 4: Translation quality results (uncased BLEU-4 %). Per-epoch times are in minutes (Min.). Statistical significance relative to D+SPARSE, the strongest baseline: **bold** ($p < 0.001$) and ***bold-italic*** ($p < 0.05$). Significance is computed by the permutation test of Riezler and Maxwell (2005). †The dev score of EXT+FILT+DOM is the dev-dom data set from Table 3, so it is not comparable with the other rows.

EXT+FILT+DOM to the baselines and other feature sets. The gains relative to SPARSE are statistically significant for all six test sets.

A crucial result is that with domain features accuracy relative to EXT+FILT never decreases: a single domain-adapted system is effective across domains. Irvine et al. (2013) showed that when models from multiple domains are interpolated, scoring errors affecting lexical selection—the model could have generated the correct target lexical item but did not—increase significantly. We do not observe that behavior, at least from the perspective of BLEU.

Table 5 separates out per-domain results. The web data appears to be the hardest domain. That is sensible given that broadcast news transcripts are more similar to newswire, the default domain, than web data. Moreover, inspection of the bitext sources revealed very little web data, so our automatic data selection is probably less effective. Accuracy on newswire actually increases slightly.

Ar-En	test1		test2		test3		
	nw	wb	nw	wb	bn	nw	wb
EF	59.78	39.55	51.69	**38.80**	30.39	37.59	20.58
EFD	**60.21**	**40.38**	**51.76**	38.77	**31.63**	**38.18**	**22.37**
Zh-En							
EF	34.56	**21.94**	17.38	12.07	**3.04**	17.42	12.83
EFD	**34.87**	21.82	**17.96**	**12.66**	3.01	**17.74**	**13.80**

Table 5: Per-domain results (uncased BLEU-4 %). Here **bold** simply indicates the maximum in each column. Model abbreviations: EF is EXT+FILT and EFD is EXT+FILT+DOM.

6 Analysis

6.1 Learning

Loss Function In a now classic empirical comparison of batch tuning algorithms, Cherry and Foster (2012) showed that PRO and expected BLEU yielded similar translation quality results. In contrast, Table 6a shows significant differences between these loss functions. First, expected BLEU can be computed faster since it is linear in the n-best list size, whereas exact computation of the PRO objective is $O(n^2)$ (thus sampling is often used). It also converges faster. Second, PRO tends to select larger models.[10] Finally, PRO seems to overfit on the tuning set, since there are no gains on test1.

Feature Selection A common yet crude method of feature selection is frequency cutoffs on the

[10]PRO L_1 regularization strength of $\lambda = 0.01$, above which model size decreases but translation quality degrades.

Loss	#epochs	Min./Epoch	#feat.	tune	test1
EB	17	14	94,606	59.13	50.02
PRO	14	25	181,542	61.20	50.09

(a) PRO vs. expected BLEU (EB) for Ext+Filt.

Feature Selection	#features	tune	test1
L_1	94,606	59.13	50.02
Freq. cutoffs	23,617	56.84	49.79

(b) Feature selection for Ext+Filt.

Model	#refs	tune	test1
Dense	4	49.52	47.98
Dense	1	49.34	47.78
Ext+Filt	4	59.13	50.02
Ext+Filt	1	55.39	48.88

(c) Single- vs. multiple-reference tuning.

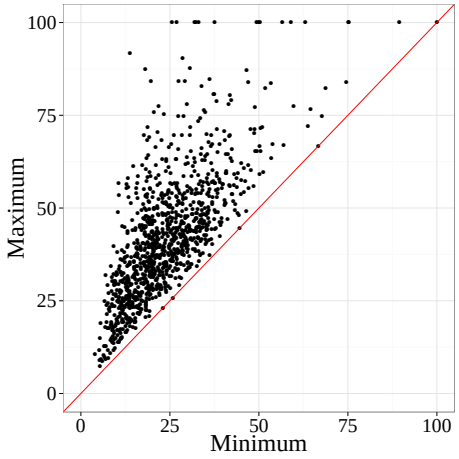
Table 6: Ar-En learning comparisons.

tuning data. Only features that fire more than some threshold are admitted into the feature set. Table 6b shows that for our new feature set, L_1 regularization—which simply requires setting a regularization strength parameter—is more effective than frequency cutoffs.

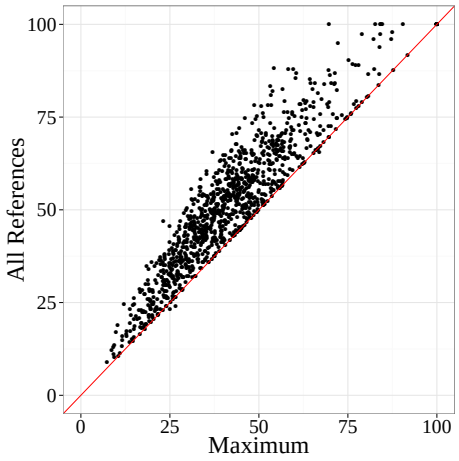
References Few MT data sets supply multiple references. Even when they do, those references are but a sample from a larger pool of possible translations. This observation has motivated attempts at generating lattices of translations for evaluation (Dreyer and Marcu, 2012; Bojar et al., 2013). But evaluation is only part of the problem. Table 6c shows that the Dense model, which has only a few features to describe the data, is little affected by the elimination of references. In contrast, the feature-rich model degrades significantly. This may account for the underperformance of features in single-reference settings like WMT (Durrani et al., 2013; Green et al., 2013a). The next section explores the impact of references further.

6.2 Reference Variance

We took the Dense Ar-En output for the dev data, which has four references, and computed the sentence-level BLEU+1 with respect to each reference. Figure 1a shows a point for each of the 1,075 translations. The horizontal axis is the minimum score with respect to any reference and the vertical axis is the maximum (BLEU has a maximum value of 1.0). Ideally, from the perspective of learning, the scores should cluster around the diagonal: the references should yield similar scores. This is hardly the case. The mean difference is $M = 18.1$ BLEU, with a standard deviation $SD = 11.5$.



(a) Maximum vs. minimum BLEU+1 (%)



(b) BLEU+1 (%) according to all four references vs. maximum

Figure 1: Reference choice analysis for Ar-En Dense output on the dev set.

Figure 1b shows the same data set, but with the maximum on the horizontal axis and the multiple-reference score on the vertical axis. Assuming a constant brevity penalty, the maximum lower-bounds the multiple-reference score since BLEU aggregates n-grams across references. The multiple-reference score is an "easier" target since the model has more opportunities to match n-grams.

Consider again the single-reference condition and one of the pathological cases at the top of Figure 1a. Suppose that the low-scoring reference is observed in the single-reference condition. The more expressive feature-rich model has a greater capacity to fit that reference when, under another

reference, it would have matched the translation exactly and incurred a low loss.

Nakov et al. (2012) suggested extensions to BLEU+1 that were subsequently found to improve accuracy in the single-reference condition (Gimpel and Smith, 2012a). Repeating the min/max calculations with the most effective extensions (according to Gimpel and Smith (2012a)) we observe lower variance ($M = 17.32$, $SD = 10.68$). These extensions are very simple, so a more sophisticated noise model is a promising future direction.

7 Related Work

We review work on phrase-based discriminative feature sets that influence decoder search, and domain adaptation with features.[11]

7.1 Feature Sets

Variants of some extended features are scattered throughout previous work: unfiltered lexicalized rule indicators and alignments (Liang et al., 2006a); rule shape (Hopkins and May, 2011); rule orientation (Liang et al., 2006b; Cherry, 2013); target unigram class (Ammar et al., 2013). We found that other prior features did not improve translation: higher-order target lexical n-grams (Liang et al., 2006a; Watanabe et al., 2007; Gimpel and Smith, 2012b), higher-order target class n-grams (Ammar et al., 2013), target word insertion (Watanabe et al., 2007; Chiang et al., 2009), and many other unpublished ideas transmitted through received wisdom.

To our knowledge, Yu et al. (2013) were the first to experiment with non-local (derivation) features for phrase-based MT. They added discriminative rule features conditioned on target context. This is a good idea that we plan to explore. However, they do not mention if their non-local features declare recombination state. Our empirical experience is that non-local features are less effective when they do not influence recombination.

Liang et al. (2006a) proposed replacing lexical items with supervised part-of-speech (POS) tags to reduce sparsity. This is a natural idea that lay dormant until recently. Ammar et al. (2013) incorporated unigram and bigram target class features. Yu et al. (2013) used word classes as backoff features to reduce overfitting. Wuebker et al. (2013) replaced all lexical items in the bitext and monolingual data with classes, and estimated the dense feature set. Then they added these dense class-based features to the baseline lexicalized system. Finally, Cherry (2013) experimented with class-based hierarchical reordering features. However, his features used a bespoke representation rather than the simple full rule string that we use.

7.2 Domain Adaptation with Features

Both Clark et al. (2012) and Wang et al. (2012) augmented the baseline dense feature set with domain labels. They each showed modest improvements for several language pairs. However, neither incorporated a notion of a default prior domain.

Liu et al. (2012) investigated local adaption of the log-linear scores by selecting comparable bitext examples for a given source input. After selecting a small local corpus, their algorithm then performs several online update steps—starting from a globally tuned weight vector—prior to decoding the input. The resulting model is effectively a locally weighted, domain-adapted classifier.

Su et al. (2012) proposed domain adaptation via monolingual source resources much as we use in-domain monolingual corpora for data selection. They labeled each bitext sentence with a topic using a Hidden Topic Markov Model (HTMM) Gruber et al. (2007). Source topic information was then mixed into the translation model dense feature calculations. This work follows Chiang et al. (2011), who present a similar technique but using the same gold NIST labels that we use. Hasler et al. (2012) extended these ideas to a discriminative sparse feature set by augmenting both rule and unigram alignment features with HTMM topic information.

8 Conclusion

This paper makes four major contributions. First, we introduced *extended features* for phrase-based MT that exceeded both dense and feature-rich baselines. Second, we specialized the features to source domains, further extending the gains. Third, we showed that online expected BLEU is faster and more stable than online PRO for extended features. Finally, we released fast, scalable, language-independent tools for implementing the feature set. Our work should help practitioners quickly establish higher baselines on the way to more targeted linguistic features. However, our analysis showed that reference choice may restrain otherwise justifiable enthusiasm for feature-rich MT.

[11] Space limitations preclude discussion of re-ranking features.

Acknowledgments We thank John DeNero for comments on an earlier version of this work. The first author is supported by a National Science Foundation Graduate Research Fellowship. 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

W. Ammar, V. Chahuneau, M. Denkowski, G. Hanneman, W. Ling, A. Matthews, et al. 2013. The CMU machine translation systems at WMT 2013: Syntax, synthetic translation options, and pseudo-references. In *WMT*.

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

O. Bojar, M. Macháček, A. Tamchyna, and D. Zeman. 2013. Scratching the surface of possible translations. In I. Habernal and V. Matoušek, editors, *Text, Speech, and Dialogue*, volume 8082 of *Lecture Notes in Computer Science*, pages 465–474. Springer Berlin Heidelberg.

P-C. Chang, M. Galley, and C. D. Manning. 2008. Optimizing Chinese word segmentation for machine translation performance. In *WMT*.

C. Cherry and G. Foster. 2012. Batch tuning strategies for statistical machine translation. In *HLT-NAACL*.

C. Cherry. 2013. Improved reordering for phrase-based translation using sparse features. In *HLT-NAACL*.

D. Chiang, K. Knight, and W. Wang. 2009. 11,001 new features for statistical machine translation. In *HLT-NAACL*.

D. Chiang, S. DeNeefe, and M. Pust. 2011. Two easy improvements to lexical weighting. In *ACL*.

C. Christodoulopoulos, S. Goldwater, and M. Steedman. 2010. Two decades of unsupervised POS induction: How far have we come? In *EMNLP*.

J. H. Clark, A. Lavie, and C. Dyer. 2012. One system, many domains: Open-domain statistical machine translation via feature augmentation. In *AMTA*.

H. Daumé III. 2007. Frustratingly easy domain adaptation. In *ACL*.

M. Dreyer and D. Marcu. 2012. HyTER: Meaning-equivalent semantics for translation evaluation. In *NAACL*.

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.

N. Durrani, B. Haddow, K. Heafield, and P. Koehn. 2013. Edinburgh's machine translation systems for European language pairs. In *WMT*.

J. R. Finkel and C. D. Manning. 2009. Hierarchical bayesian domain adaptation. In *HLT-NAACL*.

J. Gao and X. He. 2013. Training MRF-based phrase translation models using gradient ascent. In *NAACL*.

K. Gimpel and N. A. Smith. 2012a. Addendum to structured ramp loss minimization for machine translation. Technical report, Language Technologies Institute, Carnegie Mellon University.

K. Gimpel and N. A. Smith. 2012b. Structured ramp loss minimization for machine translation. In *HLT-NAACL*.

S. Green, D. Cer, K. Reschke, R. Voigt, J. Bauer, S. Wang, and others. 2013a. Feature-rich phrase-based translation: Stanford University's submission to the WMT 2013 translation task. In *WMT*.

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

S. Green, D. Cer, and C. D. Manning. 2014. Phrasal: A toolkit for new directions in statistical machine translation. In *WMT*.

A. Gruber, Y. Weiss, and M. Rosen-Zvi. 2007. Hidden topic markov models. In *AISTATS*.

E. Hasler, B. Haddow, and P. Koehn. 2012. Sparse lexicalised features and topic adaptation for SMT. In *IWSLT*.

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

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

A. Irvine, J. Morgan, M. Carpuat, H. Daumé III, and D. Munteanu. 2013. Measuring machine translation errors in new domains. *TACL*, 1.

P. Koehn, F. J. Och, and D. Marcu. 2003. Statistical phrase-based translation. In *NAACL*.

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, A. Bouchard-Côté, D. Klein, and B. Taskar. 2006a. An end-to-end discriminative approach to machine translation. In *ACL*.

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

P. Liang. 2005. Semi-supervised learning for natural language. Master's thesis, Massachusetts Institute of Technology.

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

L. Liu, H. Cao, T. Watanabe, T. Zhao, M. Yu, and C. Zhu. 2012. Locally training the log-linear model for SMT. In *EMNLP-CoNLL*.

M. Maamouri, A. Bies, and S. Kulick. 2008. Enhancing the Arabic Treebank: A collaborative effort toward new annotation guidelines. In *LREC*.

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

W. Monroe, S. Green, and C. D. Manning. 2014. Word segmentation of informal Arabic with domain adaptation. In *ACL, Short Papers*.

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

P. Nakov, F. Guzmán, and S. Vogel. 2013. A tale about PRO and monsters. In *ACL, Short Papers*.

F. J. Och and H. Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

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*.

K. Papineni, S. Roukos, T. Ward, and W. Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *ACL*.

S. Riezler and J. T. Maxwell. 2005. On some pitfalls in automatic evaluation and significance testing in MT. In *ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*.

G. Salton. 1958. The use of punctuation patterns in machine translation. *Mechanical Translation*, 5(1):16–24, July.

J. Su, H. Wu, H. Wang, Y. Chen, X. Shi, H. Dong, and Q. Liu. 2012. Translation model adaptation for statistical machine translation with monolingual topic information. In *ACL*.

J. Uszkoreit and T. Brants. 2008. Distributed word clustering for large scale class-based language modeling in machine translation. In *ACL-HLT*.

W. Wang, K. Macherey, W. Macherey, F. J. Och, and P. Xu. 2012. Improved domain adaptation for statistical machine translation. In *AMTA*.

T. Watanabe, J. Suzuki, H. Tsukada, and H. Isozaki. 2007. Online large-margin training for statistical machine translation. In *EMNLP-CoNLL*.

E. W. D. Whittaker and P. C. Woodland. 2001. Efficient class-based language modelling for very large vocabularies. In *ICASSP*.

J. Wuebker, S. Peitz, F. Rietig, and H. Ney. 2013. Improving statistical machine translation with word class models. In *EMNLP*.

N. Xue, F. Xia, F. Chiou, and M. Palmer. 2005. The Penn Chinese Treebank: Phrase structure annotation of a large corpus. *Natural Language Engineering*, 11(2):207–238.

H. Yu, L. Huang, H. Mi, and K. Zhao. 2013. Max-violation perceptron and forced decoding for scalable MT training. In *EMNLP*.