Rich Source-Side Context for Statistical Machine Translation

Kevin Gimpel and **Noah A. Smith**
Language Technologies Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213, USA
{kgimpel,nasmith}@cs.cmu.edu

Abstract

We explore the augmentation of statistical machine translation models with features of the *context* of each phrase to be translated. This work extends several existing threads of research in statistical MT, including the use of context in example-based machine translation (Carl and Way, 2003) and the incorporation of word sense disambiguation into a translation model (Chan et al., 2007). The context features we consider use surrounding words and part-of-speech tags, local syntactic structure, and other properties of the source language sentence to help predict each phrase's translation. Our approach requires very little computation beyond the standard phrase extraction algorithm and scales well to large data scenarios. We report significant improvements in automatic evaluation scores for Chinese-to-English and English-to-German translation, and also describe our entry in the WMT08 shared task based on this approach.

1 Introduction

Machine translation (MT) by statistical modeling of bilingual phrases is one of the most successful approaches in the past few years. Phrase-based MT systems are straightforward to train from parallel corpora (Koehn et al., 2003) and, like the original IBM models (Brown et al., 1990), benefit from standard language models built on large monolingual, target-language corpora (Brants et al., 2007). Many of these systems perform well in competitive evaluations and scale well to large-data situations (NIST, 2006; Callison-Burch et al., 2007). End-to-end phrase-based MT systems can be built entirely from freely-available tools (Koehn et al., 2007).

We follow the approach of Koehn et al. (2003), in which we translate a source-language sentence $\boldsymbol{f}$ into the target-language sentence $\hat{\boldsymbol{e}}$ that maximizes a linear combination of features and weights:[1]

$$\langle \hat{\boldsymbol{e}}, \hat{\boldsymbol{a}} \rangle = \underset{\langle \boldsymbol{e}, \boldsymbol{a} \rangle}{\operatorname{argmax}} \operatorname{score}(\boldsymbol{e}, \boldsymbol{a}, \boldsymbol{f}) \quad (1)$$

$$= \underset{\langle \boldsymbol{e}, \boldsymbol{a} \rangle}{\operatorname{argmax}} \sum_{m=1}^{M} \lambda_m h_m(\boldsymbol{e}, \boldsymbol{a}, \boldsymbol{f}) \quad (2)$$

where $\boldsymbol{a}$ represents the segmentation of $\boldsymbol{e}$ and $\boldsymbol{f}$ into phrases and a correspondence between phrases, and each h_m is a $\mathbb{R}$-valued feature with learned weight λ_m. The translation is typically found using beam search (Koehn et al., 2003). The weights $\langle \lambda_1, ..., \lambda_M \rangle$ are typically learned to directly minimize a standard evaluation criterion on development data (e.g., the BLEU score; Papineni et al., (2002)) using numerical search (Och, 2003).

Many features are used in phrase-based MT, but nearly ubiquitous are estimates of the conditional translation probabilities $p(\boldsymbol{e}_i^j \mid \boldsymbol{f}_k^\ell)$ and $p(\boldsymbol{f}_k^\ell \mid \boldsymbol{e}_i^j)$ for each phrase pair $\langle \boldsymbol{e}_i^j, \boldsymbol{f}_k^l \rangle$ in the candidate sentence pair.[2] In this paper, we add and evaluate fea-

[1] In the statistical MT literature, this is often referred to as a "log-linear model," but since the score is normalized during neither parameter training nor decoding, and is never interpreted as a log-probability, it is essentially a linear combination of feature functions. Since many of the features are actually probabilities, this linear combination is closer to a *mixture model*.

[2] We will use $\boldsymbol{x}_i^j$ to denote the subsequence of $\boldsymbol{x}$ containing the ith through jth elements of $\boldsymbol{x}$, inclusive.

Proceedings of the Third Workshop on Statistical Machine Translation, pages 9–17,
Columbus, June 2008.

tures that condition on additional context features on the *source* ($\boldsymbol{f}$) side:

$$p(\boldsymbol{e}_i^j \mid Phrase = \boldsymbol{f}_k^\ell, Context = \langle \boldsymbol{f}_1^{k-1}, \boldsymbol{f}_{\ell+1}^F, ...\rangle)$$

The advantage of considering context is well-known and exploited in the example-based MT community (Carl and Way, 2003). Recently researchers have begun to use source phrase context information in statistical MT systems (Stroppa et al., 2007). Statistical NLP researchers understand that conditioning a probability model on more information is helpful only if there are sufficient training data to accurately *estimate* the context probabilities.[3] Sparse data are often the death of elaborate models, though this can be remedied through careful smoothing.

In this paper we leverage the existing linear model (Equation 2) to bring source-side context into phrase-based MT in a way that is robust to data sparseness. We interpret the linear model as a mixture of many probability estimates based on different context features, some of which may be very sparse. The mixture coefficients are trained in the usual way ("minimum error-rate training," Och, 2003), so that the additional context is exploited when it is useful and ignored when it isn't.

The paper proceeds as follows. We first review related work that enriches statistical translation models using context (§2). We then propose a set of source-side features to be incorporated into the translation model, including the novel use of syntactic context from source-side parse trees and global position within $\boldsymbol{f}$ (§3). We explain why analogous *target*-side features pose a computational challenge (§4). Specific modifications to the standard training and evaluation paradigm are presented in §5. Experimental results are reported in §6.

2 Related Work

Stroppa et al. (2007) added souce-side context features to a phrase-based translation system, including conditional probabilities of the same form that we use. They consider up to two words and/or POS tags of context on either side. Because of the aforementioned data sparseness problem, they use a decision-tree classifier that implicitly smooths relative frequency estimates. The method improved over a standard phrase-based baseline trained on small amounts of data (< 50K sentence pairs) for Italian → English and Chinese → English. We explore a significantly larger space of context features, a smoothing method that more naturally fits into the widely used, error-driven linear model, and report a more comprehensive experimental evaluation (including feature comparison and scaling up to very large datasets).

Recent research on the use of word-sense disambiguation in machine translation also points toward our approach. For example, Vickrey et al. (2005) built classifiers inspired by those used in word sense disambiguation to fill in blanks in a partially-completed translation. Giménez and Màrquez (2007) extended the work by considering phrases and moved to full translation instead of filling in target-side blanks. They trained an SVM for each source language phrase using local features of the sentences in which the phrases appear. Carpuat and Wu (2007) and Chan et al. (2007) embedded state-of-the-art word sense disambiguation modules into statistical MT systems, achieving performance improvements under several automatic measures for Chinese → English translation.

Our approach is also reminiscent of example-based machine translation (Nagao, 1984; Somers, 1999; Carl and Way, 2003), which has for many years emphasized use of the context in which source phrases appear when translating them. Indeed, like the example-based community, we do not begin with any set of assumptions about *which kinds* of phrases require additional disambiguation (cf. the application of word-sense disambiguation, which is motivated by lexical ambiguity). Our feature-rich approach is omnivorous and can exploit *any* linguistic analysis of an input sentence.

3 Source-Side Context Features

Adding features to the linear model (Equation 2) that consider more of the source sentence requires changing the decoder very little, if at all. The reason is that the source sentence is fully observed, so the information to be predicted is the same as before—the difference is that we are using more clues to carry out the prediction.

[3] An illustrative example is the debate over the use of bilexicalized grammar rules in statistical parsing (Gildea, 2001; Bikel, 2004).

We see this as an opportunity to include many more features in phrase-based MT without increasing the cost of decoding at runtime. This discussion is reminiscent of an advantage gained by moving from hidden Markov models to conditional random fields for sequence labeling tasks. While the same core algorithm is used for decoding with both models, a CRF allows inclusion of features that consider the *entire* observed sequence—i.e., more of the observable context of each label to be predicted. Although this same advantage was already obtained in statistical MT through the transition from "noisy channel" translation models to (log-)linear models, the customary set of features used in most phrase-based systems does not take full advantage of the observed data.

The standard approach to estimating the phrase translation conditional probability features is via relative frequencies (here $\boldsymbol{e}$ and $\boldsymbol{f}$ are phrases):

$$p(\boldsymbol{e} \mid \boldsymbol{f}) = \frac{count(\boldsymbol{e}, \boldsymbol{f})}{\sum_{\boldsymbol{e'}} count(\boldsymbol{e'}, \boldsymbol{f})}$$

Our new features all take the form $p(\boldsymbol{e} \mid \boldsymbol{f}, \boldsymbol{f}_{context})$, where $\boldsymbol{e}$ is the target language phrase, $\boldsymbol{f}$ is the source language phrase, and $\boldsymbol{f}_{context}$ is the context of the source language phrase in the sentence in which it was observed. Like the context-bare conditional probabilities, we estimate probability features using relative frequencies:

$$p(\boldsymbol{e} \mid \boldsymbol{f}, \boldsymbol{f}_{context}) = \frac{count(\boldsymbol{e}, \boldsymbol{f}, \boldsymbol{f}_{context})}{\sum_{\boldsymbol{e'}} count(\boldsymbol{e'}, \boldsymbol{f}, \boldsymbol{f}_{context})}$$

Since we expect that adding conditioning variables will lead to sparser counts and therefore more zero estimates, we compute features for many different types of context. To combine the many differently-conditioned features into a single model, we provide them as features to the linear model (Equation 2) and use minimum error-rate training (Och, 2003) to obtain interpolation weights λ_m. This is similar to an interpolation of backed-off estimates, if we imagine that all of the different contexts are differently-backed off estimates of the *complete* context. The error-driven weight training effectively smooths one implicit context-rich estimate $p(\boldsymbol{e} \mid \boldsymbol{f}, \boldsymbol{f}_{context})$ so that all of the backed-off estimates are taken into account, including the original $p(\boldsymbol{e} \mid \boldsymbol{f})$. Our approach is asymmetrical; we have not, for example, estimated features of the form $p(\boldsymbol{f}, \boldsymbol{f}_{context} \mid \boldsymbol{e})$.

We next discuss the specific source-side context features used in our model.

3.1 Lexical Context Features

The most obvious kind of context of a source phrase $\boldsymbol{f}_k^\ell$ is the m-length sequence before it ($\boldsymbol{f}_{k-m}^{k-1}$) and the m-length sequence after it ($\boldsymbol{f}_{\ell+1}^{\ell+m}$). We include context features for $m \in \{1, 2\}$, padding sentences with m special symbols at the beginning and at the end. For each value of m, we include three features:

- $p(\boldsymbol{e} \mid \boldsymbol{f}, \boldsymbol{f}_{k-m}^{k-1})$, the left lexical context;
- $p(\boldsymbol{e} \mid \boldsymbol{f}, \boldsymbol{f}_{\ell+1}^{\ell+m})$, the right lexical context;
- $p(\boldsymbol{e} \mid \boldsymbol{f}, \boldsymbol{f}_{k-m}^{k-1}, \boldsymbol{f}_{\ell+1}^{\ell+m})$, both sides.

3.2 Shallow Syntactic Features

Lexical context features, especially when $m > 1$, are expected to be sparse. Representing the context by part-of-speech (POS) tags is one way to overcome that sparseness. We used the same set of the lexical context features described above, but with POS tags replacing words in the context. We also include a feature which conditions on the POS tag sequence of the actual phrase being translated.

3.3 Syntactic Features

If a robust parser is available for the source language, we can include context features from parse trees. We used the following parse tree features:

- Is the phrase (exactly) a constituent?
- What is the nonterminal label of the lowest node in the parse tree that covers the phrase?
- What is the nonterminal label or POS of the highest nonterminal node that ends immediately before the phrase? Begins immediately after the phrase?
- Is the phrase strictly to the left of the root word, does it contain the root word, or is it strictly to the right of the root word? (Requires a parse with head annotations.)

We also used a feature that conditions on both features in the third bullet point above.

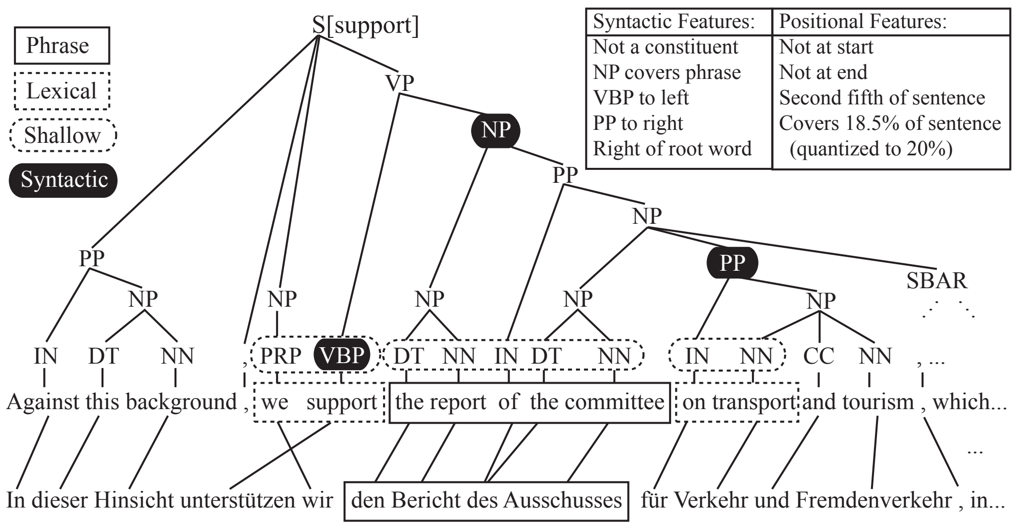

Figure 1: A (partial) sentence pair from the WMT07 Europarl training corpus. Processing of the data (parsing, word alignment) was done as discussed in §6. The phrase pair of interest is boxed and context features are shown in dotted shapes. The context features help determine whether the phrase should be translated as "der Bericht des Ausschusses" (nominative case) or "den Bericht des Ausschusses" (accusative case). See text for details.

3.4 Positional Features

We include features based on the position of the phrase in the source sentence, the phrase length, and the sentence length. These features use information from the entire source sentence, but are not syntactic. For a phrase $\boldsymbol{f}_k^\ell$ in a sentence $\boldsymbol{f}$ of length n:

- Is the phrase at the start of the sentence ($k = 1$)?
- Is the phrase at the end of the sentence ($\ell = n$)?
- A quantization of $r = \frac{k+\frac{\ell-k+1}{2}}{n}$, the relative position in $(0, 1)$ of the phrase's midpoint within $\boldsymbol{f}$. We choose the smallest $q \in \{0.2, 0.4, 0.6, 0.8, 1\}$ such that $q > r$.
- A quantization of $c = \frac{\ell-k+1}{n}$, the fraction of the words in $\boldsymbol{f}$ that are covered by the phrase. We choose the smallest $q \in \{\frac{1}{40}, \frac{1}{20} \cdot \frac{1}{10}, \frac{1}{5}, \frac{1}{3}, 1\}$ such that $q > c$.

An illustration of the context features is shown in Fig. 1. Consider the phrase pair "the report of the committee"/"den Bericht des Ausschusses" extracted by our English → German baseline MT system (described in §6.3). The German word "Bericht" is a masculine noun; therefore, it takes the article "der" in the nominative case, "den" in the accusative case, and "dem" in the dative case. These three translations are indeed available in the phrase table for "the report of the committee" (see Table 1, "no context" column), with relatively high entropy. The choice between "den" and "der" must be made by the language model alone.

Knowing that the phrase follows a verb, or appears to the right of the sentence's root word, or within the second fifth of the sentence should help. Indeed, a probability distribution that conditions on context features gives more peaked distributions that give higher probability to the correct translation, given *this* context, and lower probability given some *other* contexts (see Table 1).

4 Why Not Target-Side Context?

While source context is straightforward to exploit in a model, including target-side context features breaks one of the key independence assumptions made by phrase-based translation models: the translations of the source-side phrases are conditionally *independent* of each other, given $\boldsymbol{f}$, thereby requiring new algorithms for decoding (Marino et al., 2006).

We suggest that target-side context may already be well accounted for in current MT systems. Indeed, *language models* pay attention to the local context of phrases, as do reordering models. The recent emphasis on improving these components of a translation system (Brants et al., 2007) is likely due in part to the widespread availability of NLP tools for the language that is most frequently the target: English. We will demonstrate that NLP tools (tag-

$\boldsymbol{g}$	no context	Shallow: 2 POS on left *"PRP VBP"	"VBN IN"	Syntax: _ of root *right	left	Positional: rel. pos. *2nd fifth	1st fifth
den bericht des ausschusses	0.3125	**1.0000**	0.3333	**0.5000**	0.0000	**0.6000**	0.0000
der bericht des ausschusses	0.3125	0.0000	0.0000	0.1000	**0.6667**	0.2000	**0.6667**
dem bericht des ausschusses	0.2500	0.0000	**0.6667**	0.3000	0.1667	0.0000	0.1667

Table 1: Phrase table entries for "the report of the committee" and their scores under different contexts. These are the top three phrases in the baseline English → German system ("no context" column). Contexts from the source sentence in Fig. 1 (starred) predict correctly; we show also alternative contexts that give very different distributions.

gers and parsers) for the *source* side can be used to improve the translation model, exploiting analysis tools for other languages.

5 Implementation

The additional data required to compute the context features is extracted along with the phrase pairs during execution of the standard phrase extraction algorithm, affecting phrase extraction and scoring time by a constant factor.

We avoid the need to modify the standard phrase-based decoder to handle context features by appending a unique identifier to each token in the sentences to be translated. Then, we pre-compute a phrase table for the phrases in these sentences according to the phrase contexts. To avoid extremely long lists of translations of common tokens, we filter the generated phrase tables, removing entries for which the estimate of $p(\boldsymbol{e} \mid \boldsymbol{f}) < c$, for some small c. In our experiments, we fixed $c = 0.0002$. This filtering reduced time for experimentation dramatically and had no apparent effect on the translation output. We did not perform any filtering for the baseline system.

6 Experiments

In this section we present experimental results using our context-endowed phrase translation model with a variety of different context features, on Chinese → English, German → English, and English → German translation tasks. Dataset details are given in Appendices A (Chinese) and B (German).

Context features	Chinese → English (UN) BLEU	NIST	METEOR
None	0.3426	7.740	0.6416
Lexical	**0.3678**	**8.107**	**0.6627**
Shallow	**0.3473**	7.724	**0.6452**
Lexical + Shallow	**0.3669**	**8.117**	**0.6609**
Syntactic	**0.3523**	**7.791**	**0.6481**
Positional	**0.3480**	7.764	**0.6446**
All	**0.3620**	**7.953**	**0.6570**

Table 2: Chinese → English experiments: training and testing on unseen UN data. Boldface marks scores signficantly higher than "None."

Baseline We use the Moses MT system (Koehn et al., 2007) as a baseline and closely follow the example training procedure given for the WMT2007 and 2008 shared tasks.[4] In particular, we perform word alignment in each direction using GIZA++ (Och and Ney, 2003), apply the "grow-diag-final-and" heuristic for symmetrization with maximum phrase length of 7. In addition to the two phrase translation conditionals $p(\boldsymbol{e} \mid \boldsymbol{f})$ and $p(\boldsymbol{f} \mid \boldsymbol{e})$, we use lexical translation probabilities in each direction, a word penalty, a phrase penalty, a length-based reordering model, a lexicalized reordering model, and an n-gram language model, SRILM implementation (Stolcke, 2002) with modified Kneser-Ney smoothing (Chen and Goodman, 1998). Minimum error-rate (MER) training (Och, 2003) was applied to obtain weights (λ_m in Equation 2) for these features. A recaser is trained on the target side of the parallel corpus using the script provided with Moses. All output is recased and detokenized prior to evaluation.

Evaluation We evaluate translation output using three automatic evaluation measures: BLEU (Papineni et al., 2002), NIST (Doddington, 2002), and METEOR (Banerjee and Lavie, 2005, version 0.6).[5] All measures used were the case-sensitive, corpus-level versions. The version of BLEU used was that provided by NIST and was also used as the evaluation measure for MER training. Significance was tested using a paired bootstrap (Koehn, 2004) with

[4] `http://www.statmt.org/wmt08`

[5] METEOR details: For English, we use exact matching, Porter stemming, and WordNet synonym matching. For German, we use exact matching and Porter stemming. These are the same settings that were used to evaluate systems for the WMT07 shared task.

Context features	Chinese → English Testing on UN BLEU	NIST	METEOR	Testing on News (NIST 2003) BLEU	NIST	METEOR
Training on in-domain data only:						
None	0.3426	7.740	0.6416	0.2686	8.085	0.5346
Training on all data:						
None	0.3347	7.618	0.6352	0.2663	7.800	0.5213
Lexical	**0.3543**	**7.942**	**0.6612**	0.2580	7.867	0.5242
Shallow: ≤ 1 POS tag	0.3279	7.427	0.6380	0.2683	**8.361**	**0.5471**
Shallow: ≤ 2 POS tags	0.3341	7.529	0.6403	0.2654	7.937	0.5263
Lexical + Shallow	**0.3535**	**7.965**	**0.6584**	0.2691	7.917	0.5276
Syntactic	0.3424	7.704	**0.6483**	0.2640	**8.198**	**0.5390**
Lexical + Syntactic	**0.3565**	**7.916**	**0.6574**	0.2626	7.776	0.5205
Positional	0.3300	7.473	0.6385	0.2682	7.869	0.5252
All	0.3457	**7.932**	**0.6550**	0.2641	7.793	0.5224
Feature selection (see Sec. 6.4)	**0.3536**	**7.878**	**0.6525**	**0.2779**	8.147	0.5330

Table 3: Chinese → English experiments: first row shows baseline performance when training only on in-domain data for each task; all other rows show results when training on *all* data (UN and News). Left half shows results when tuning and testing on UN test sets; right half shows results when tuning on NIST 2004 News test set and testing on NIST 2003. Boldface marks scores that are significantly higher than the first row, in-domain baseline.

1000 samples ($p < 0.05$).[6]

6.1 Chinese → English

For our Chinese → English experiments, two kinds of data were used: UN proceedings, and newswire as used in NIST evaluations.

UN Data UN data results are reported in Table 2. Significant improvements are obtained on all three evaluation measures—e.g., more than 2 BLEU points—using lexical or lexical and shallow features. While improvements are smaller for other features and feature combinations, performance is not *harmed* by conditioning on context features, with one very minor exception (shallow features slighly harm the NIST score). Note that using syntactic features gave 1 BLEU point of improvement.

News Data In News data experiments, none of our features obtained BLEU performance statistically distinguishable from the baseline of 0.2686 BLEU (neither better, nor worse). The News training corpus is less than half the size of the UN training corpus (in words); unsurprisingly, the context features were too sparse to be helpful. Further, newswire are less formulaic and repetitive than UN proceedings, so contexts do not generalize as well from training to test data. Fortunately, our "error-minimizing mixture" approach protects the BLEU score, which the λ_m are tuned to optimize.

Combined UN + News Data Our next experiment used *all* of the available training data (> 200M words on each side) to train the models, in-domain λ_m tuning, and testing for each domain separately; see Table 3. Without context features, training on mixed-domain data consistently *harms* performance. With contexts that include lexical features, the mixed-domain model significantly outperforms the *in-domain* baseline for UN data. These results suggest that context features enable better use of out-of-domain data, an important advantage for statistical MT since parallel data often arise from very different sources than those of "real-world" translation scenarios. On News data, context features did not give a significant advantage on the BLEU score, though syntactic and ≤ 1 POS contexts did give significant NIST and METEOR improvements over the in-domain baseline. Small sets of automatically selected context features, discussed in Section 6.4, were more consistently successful for these data.

6.2 German → English

We do not report full results for this task, because the context features neither helped nor hurt performance significantly. We believe this is due to data

[6] Code implementing this test for these metrics can be freely downloaded at `http://www.ark.cs.cmu.edu/MT`.

	English → German		
Context features	BLEU	NIST	METEOR
None	0.2018	5.874	0.2753
Lexical	0.1958	5.884	0.2703
Shallow	0.1989	5.833	0.2731
Syntactic	0.2024	**5.945**	**0.2777**
Positional	0.2008	5.860	0.2733
Lex. + Shal. + Syn.	0.2000	**5.959**	0.2764
All	0.1996	5.868	0.2738
Feature selection	**0.2055**	**5.939**	**0.2778**

Table 4: English → German experiments: training and testing on Europarl data. Boldface marks scores significantly higher than "None."

sparseness resulting from the size of the training corpus (26M German words), German's relatively rich morphology, and the challenges of German parsing.

6.3 English → German

English → German results are shown in Table 4. The baseline system here is highly competitive, having scored higher on automatic evaluation measures than any other system in the WMT07 shared task (Callison-Burch et al., 2007). Among the feature categories, the largest improvement is achieved when *syntactic* context features are included. Comparing with the German → English experiment, we attribute this effect to the high accuracy of the English parser compared to the German parser.

6.4 Feature Selection

Translation performance does not always increase when features are added to the model. This motivates the use of feature selection algorithms to choose a subset of features to optimize performance. We experimented with several feature selection algorithms based on information-theoretic quantities computed among the source phrase, the target phrase, and the context, but found that a simple forward variable selection algorithm (Guyon and Elisseeff, 2003) worked best. In this procedure, we start with no context features and, at each iteration, add the single feature that results in the largest increase in BLEU score on the unseen development test data after λ_m tuning. The algorithm terminates if no features are left or if none result in an increase in BLEU. We ran this algorithm to completion for the two Chinese → English tune/test sets (training on *all* data in each case) and the English → German task; see Tables 3 and 4. In all cases, the algorithm finishes after ≤ 4 evaluations.

Feature selection for Chinese → English (UN) first chose the lexical feature "1 word on each side," then the positional feature indicating which fifth of the sentence contains the phrase, and finally the lexical feature "1 word on right." For News, the features chosen were the shallow syntactic feature "1 POS on each side," then the positional beginning-of-sentence feature, then the position relative to the root (a syntactic feature). Features selected for English → German were the shallow syntactic feature "2 POS on left," then the lexical feature "1 word on right."

This simple procedure led to the best BLEU scores for the Chinese → English News task and the English → German task, showing that only a few well-chosen context features are required to give significant improvements over the baseline zero-context model. On Chinese News, our BLEU score of 0.2779 is significantly better than the *in-domain* baseline system score of 0.2686.

6.5 WMT08 Shared Task: English → German

Since we began this research before the release of the data for the WMT08 shared task, we performed the majority of our experiments using the data released for the WMT07 shared task (see Appendix B). To prepare our entry for the 2008 shared task, we trained a baseline system on the 2008 data using a nearly identical configuration.[7] We experimented with several context feature sets, targeting the features that performed best in our earlier experiments. In addition to the devtest06 data, we translated the 2007 Europarl test set to see how our feature selection results would transfer to new data. We found the best-performing feature set from our earlier experiments to also perform competitively on the new test data; Table 5 shows results consistent with experiments above.

7 Future Work

In future work, we plan to apply more sophisticated learning algorithms to rich-feature phrase table esti-

[7] The only differences were the use of a larger max sentence length threshold of 55 tokens instead of 50, and the use of the better-performing "englishFactored" Stanford parser model.

	devtest06			test07			test08		
System	BLEU	NIST	METEOR	BLEU	NIST	METEOR	BLEU	NIST	METEOR
Baseline	0.2009	5.866	0.2719	0.2051	5.957	0.2782	0.2003	5.889	0.2720
Context	**0.2039**	**5.941**	**0.2784**	**0.2088**	**6.036**	**0.2826**	0.2016	**5.956**	**0.2772**

Table 5: English → German shared task system results using WMT08 Europarl parallel data for training, dev06 for tuning, and three test sets, including the final 2008 test set. The row labeled "Context" uses the top-performing feature set {2 POS on left, 1 word on right}. Boldface marks scores that are significantly higher than the baseline.

mation. Context features can also be used as conditioning variables in other components of translation models, including the lexicalized reordering model and the lexical translation model in the Moses MT system, or hierarchical or syntactic models (Chiang, 2005). Additional linguistic analysis (e.g., morphological disambiguation, named entity recognition, semantic role labeling) can be used to define new context features.

8 Conclusion

We have described a straightforward, scalable method for improving phrase translation models by modeling features of a phrase's source-side context. Our method allows incorporation of features from any kind of source-side annotation and barely affects the decoding algorithm. Experiments show performance rivaling or exceeding strong, state-of-the-art baselines on standard translation tasks. Automatic feature selection can be used to achieve performance gains with just two or three context features. Performance is strongest when large in-domain training sets and high-accuracy NLP tools for the source language are available.

Acknowledgments

This research was supported in part by NSF IIS-0713265, supercomputing resources provided by Yahoo!, a Google grant, and an ARCS award to the first author. We thank Abhaya Agarwal, Ashish Venugopal, and Andreas Zollmann for helpful conversations and Joy Zhang for his Chinese segmenter. We also thank the anonymous reviewers for helpful comments.

A Dataset Details (Chinese-English)

We trained on data from the NIST MT 2008 constrained Chinese-English track: Sinorama (LDC2005T10), FBIS (LDC2003E14), Hong Kong Hansards and news (LDC2004T08), Xinhua (LDC2002E18), and financial news (LDC2006E26)—total 2.5M sents., 66M Chinese words, 68M English. The newswire portion of the NIST 2004 test set and the full NIST 2003 test set were used for newswire tuning and testing, respectively (∼900 sents. each). We also used the United Nations parallel text (LDC2004E12), divided into training (4.7M sents.; words: 136M Chinese, 144M English), tuning (2K sents.), and test sets (2K sents.). We removed sentence pairs where either side was longer than 80 words, segmented all Chinese text automatically,[8] and parsed using the Stanford parser with the pre-trained "xinhuaPCFG" model (Klein and Manning, 2003). Trigram language models were trained on the English side of the parallel corpus along with approximately 115M words from the Xinhua section of the English Gigaword corpus (LDC2005T12), years 1995–2000 (total 326M words).

B Dataset Details (German-English)

For German ↔ English experiments, we used data provided for the WMT 2007 shared task (1.1M sents., 26M German words, 27M English). The Europarl tuning and development test sets from the WMT 2007 shared task were used for tuning and testing (2K sents. each). We removed sentence pairs where either side was longer than 50 words and parsed the German and English data using the Stanford parser (Klein and Manning, 2003) (with pre-trained "germanFactored" and "englishPCFG" models). 5-gram language models were trained on the entire target side of the parallel corpus (37M German words, 38M English).

[8]Available at `http://projectile.is.cs.cmu.edu/research/public/tools/segmentation/lrsegmenter/lrSegmenter.perl`.

References

S. Banerjee and A. Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proc. of ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for MT and/or Summarization*.

D. M. Bikel. 2004. A distributional analysis of a lexicalized statistical parsing model. In *Proc. of EMNLP*.

T. Brants, A. C. Popat, P. Xu, F. J. Och, and J. Dean. 2007. Large language models in machine translation. In *Proc. of EMNLP-CoNLL*.

P. F. Brown, J. Cocke, S. A. Della Pietra, V. J. Della Pietra, F. Jelinek, J. D. Lafferty, R. L. Mercer, and P. S. Roossin. 1990. A statistical approach to machine translation. *Computational Linguistics*, 16(2):79–85.

C. Callison-Burch, P. Koehn, C. Fordyce, and C. Monz, editors. 2007. *Proc. of the 2nd Workshop on Statistical Machine Translation*.

M. Carl and A. Way. 2003. *Recent Advances in Example-Based Machine Translation*. Kluwer Academic.

M. Carpuat and D. Wu. 2007. Improving statistical machine translation using word sense disambiguation. In *Proc. of EMNLP-CoNLL*.

Y. Chan, H. Ng, and D. Chiang. 2007. Word sense disambiguation improves statistical machine translation. In *Proc. of ACL*.

S. Chen and J. Goodman. 1998. An empirical study of smoothing techniques for language modeling. Technical report 10-98, Harvard University.

D. Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. In *Proc. of ACL*.

G. Doddington. 2002. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *Proc. of HLT*.

D. Gildea. 2001. Corpus variation and parser performance. In *Proc. of EMNLP*.

J. Giménez and L. Màrquez. 2007. Context-aware discriminative phrase selection for statistical machine translation. In *Proc. of the 2nd Workshop on Statistical Machine Translation*.

I. Guyon and A. Elisseeff. 2003. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182.

D. Klein and C. D. Manning. 2003. Fast exact inference with a factored model for natural language parsing. In *Advances in NIPS 15*.

P. Koehn, F. J. Och, and D. Marcu. 2003. Statistical phrase-based translation. In *Proc. of HLT-NAACL*, pages 127–133.

P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *ACL demonstration session*.

P. Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proc. of EMNLP*.

José B. Marino, Rafael E. Banchs, Josep M. Crego, Adrià de Gispert, Patrik Lambert, José A. R. Fonollosa, and Marta R. Costa-jussà. 2006. N-gram-based machine translation. *Computational Linguistics*, 32(4):527–549.

M. Nagao. 1984. A framework of a mechanical translation between Japanese and English by analogy principle. In *Proc. of the International NATO Symposium on Artificial and Human Intelligence*. Elsevier North-Holland, Inc.

NIST. 2006. NIST 2006 machine translation evaluation official results.

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

F. J. Och. 2003. Minimum error rate training for statistical machine translation. In *Proc. of ACL*, pages 160–167.

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

H. Somers. 1999. Review article: Example-based machine translation. *Machine Translation*, 14(2).

A. Stolcke. 2002. SRILM—an extensible language modeling toolkit. In *Proc. of ICSLP*.

N. Stroppa, A. van den Bosch, and A. Way. 2007. Exploiting source similarity for SMT using context-informed features. In *Proc. of TMI*.

D. Vickrey, L. Biewald, M. Teyssier, and D. Koller. 2005. Word-sense disambiguation for machine translation. In *Proc. of HLT-EMNLP*.