11,001 New Features for Statistical Machine Translation*

David Chiang and **Kevin Knight**
USC Information Sciences Institute
4676 Admiralty Way, Suite 1001
Marina del Rey, CA 90292 USA

Wei Wang
Language Weaver, Inc.
4640 Admiralty Way, Suite 1210
Marina del Rey, CA 90292 USA

Abstract

We use the Margin Infused Relaxed Algorithm of Crammer et al. to add a large number of new features to two machine translation systems: the Hiero hierarchical phrase-based translation system and our syntax-based translation system. On a large-scale Chinese-English translation task, we obtain statistically significant improvements of +1.5 BLEU and +1.1 BLEU, respectively. We analyze the impact of the new features and the performance of the learning algorithm.

1 Introduction

What linguistic features can improve statistical machine translation (MT)? This is a fundamental question for the discipline, particularly as it pertains to improving the best systems we have. Further:

- Do syntax-based translation systems have unique and effective levers to pull when designing new features?
- Can large numbers of feature weights be learned efficiently and stably on modest amounts of data?

In this paper, we address these questions by experimenting with a large number of new features. We add more than 250 features to improve a syntax-based MT system—already the highest-scoring single system in the NIST 2008 Chinese-English common-data track—by +1.1 BLEU. We also add more than 10,000 features to Hiero (Chiang, 2005) and obtain a +1.5 BLEU improvement.

*This research was supported in part by DARPA contract HR0011-06-C-0022 under subcontract to BBN Technologies.

Many of the new features use syntactic information, and in particular depend on information that is available only inside a syntax-based translation model. Thus they widen the advantage that syntax-based models have over other types of models.

The models are trained using the Margin Infused Relaxed Algorithm or MIRA (Crammer et al., 2006) instead of the standard minimum-error-rate training or MERT algorithm (Och, 2003). Our results add to a growing body of evidence (Watanabe et al., 2007; Chiang et al., 2008) that MIRA is preferable to MERT across languages and systems, even for very large-scale tasks.

2 Related Work

The work of Och et al (2004) is perhaps the best-known study of new features and their impact on translation quality. However, it had a few shortcomings. First, it used the features for reranking n-best lists of translations, rather than for decoding or forest reranking (Huang, 2008). Second, it attempted to incorporate syntax by applying off-the-shelf part-of-speech taggers and parsers to MT output, a task these tools were never designed for. By contrast, we incorporate features directly into hierarchical and syntax-based decoders.

A third difficulty with Och et al.'s study was that it used MERT, which is not an ideal vehicle for feature exploration because it is observed not to perform well with large feature sets. Others have introduced alternative discriminative training methods (Tillmann and Zhang, 2006; Liang et al., 2006; Turian et al., 2007; Blunsom et al., 2008; Macherey et al., 2008), in which a recurring challenge is scalability: to train many features, we need many train-

ing examples, and to train discriminatively, we need to search through all possible translations of each training example. Another line of research (Watanabe et al., 2007; Chiang et al., 2008) tries to squeeze as many features as possible from a relatively small dataset. We follow this approach here.

3 Systems Used

3.1 Hiero

Hiero (Chiang, 2005) is a hierarchical, string-to-string translation system. Its rules, which are extracted from unparsed, word-aligned parallel text, are synchronous CFG productions, for example:

$$\mathrm{X} \rightarrow \mathrm{X}_1 \text{ de } \mathrm{X}_2, \mathrm{X}_2 \text{ of } \mathrm{X}_1$$

As the number of nonterminals is limited to two, the grammar is equivalent to an inversion transduction grammar (Wu, 1997).

The baseline model includes 12 features whose weights are optimized using MERT. Two of the features are n-gram language models, which require intersecting the synchronous CFG with finite-state automata representing the language models. This grammar can be parsed efficiently using cube pruning (Chiang, 2007).

3.2 Syntax-based system

Our syntax-based system transforms source Chinese strings into target English syntax trees. Following previous work in statistical MT (Brown et al., 1993), we envision a noisy-channel model in which a language model generates English, and then a translation model transforms English trees into Chinese. We represent the translation model as a tree transducer (Knight and Graehl, 2005). It is obtained from bilingual text that has been word-aligned and whose English side has been syntactically parsed. From this data, we use the the GHKM minimal-rule extraction algorithm of (Galley et al., 2004) to yield rules like:

$$\text{NP-C}(x_0\text{:NPB PP(IN(of } x_1\text{:NPB))} \leftrightarrow x_1 \text{ de } x_0$$

Though this rule can be used in either direction, here we use it right-to-left (Chinese to English). We follow Galley et al. (2006) in allowing unaligned Chinese words to participate in multiple translation rules, and in collecting larger rules composed of minimal rules. These larger rules have been shown to substantially improve translation accuracy (Galley et al., 2006; DeNeefe et al., 2007).

We apply Good-Turing discounting to the transducer rule counts and obtain probability estimates:

$$P(\text{rule}) = \frac{count(\text{rule})}{count(LHS\text{-}root(\text{rule}))}$$

When we apply these probabilities to derive an English sentence e and a corresponding Chinese sentence c, we wind up with the joint probability $P(e, c)$.

The baseline model includes $\log P(e, c)$, the two n-gram language models $\log P(e)$, and other features for a total of 25. For example, there is a pair of features to punish rules that drop Chinese content words or introduce spurious English content words. All features are linearly combined and their weights are optimized using MERT.

For efficient decoding with integrated n-gram language models, all transducer rules must be binarized into rules that contain at most two variables and can be incrementally scored by the language model (Zhang et al., 2006). Then we use a CKY-style parser (Yamada and Knight, 2002; Galley et al., 2006) with cube pruning to decode new sentences.

We include two other techniques in our baseline. To get more general translation rules, we restructure our English training trees using expectation-maximization (Wang et al., 2007), and to get more specific translation rules, we relabel the trees with up to 4 specialized versions of each nonterminal symbol, again using expectation-maximization and the split/merge technique of Petrov et al. (2006).

3.3 MIRA training

We incorporate all our new features into a linear model (Och and Ney, 2002) and train them using MIRA (Crammer et al., 2006), following previous work (Watanabe et al., 2007; Chiang et al., 2008).

Let $\mathbf{e}$ stand for output strings or their derivations, and let $\mathbf{h}(\mathbf{e})$ stand for the feature vector for $\mathbf{e}$. Initialize the feature weights $\mathbf{w}$. Then, repeatedly:

- Select a batch of input sentences $\mathbf{f}_1, \ldots, \mathbf{f}_m$ and decode each $\mathbf{f}_i$ to obtain a forest of translations.
- For each i, select from the forest a set of hypothesis translations $\mathbf{e}_{i1}, \ldots, \mathbf{e}_{in}$, which are the

10-best translations according to each of:

$$\begin{gathered}\mathbf{h}(\mathbf{e}) \cdot \mathbf{w} \\ \textsc{Bleu}(\mathbf{e}) + \mathbf{h}(\mathbf{e}) \cdot \mathbf{w} \\ -\textsc{Bleu}(\mathbf{e}) + \mathbf{h}(\mathbf{e}) \cdot \mathbf{w}\end{gathered} \quad (1)$$

- For each i, select an oracle translation:

$$\mathbf{e}^* = \arg\max_{\mathbf{e}} \left(\textsc{Bleu}(\mathbf{e}) + \mathbf{h}(\mathbf{e}) \cdot \mathbf{w}\right) \quad (2)$$

  Let $\Delta\mathbf{h}_{ij} = \mathbf{h}(\mathbf{e}_i^*) - \mathbf{h}(\mathbf{e}_{ij})$.

- For each $\mathbf{e}_{ij}$, compute the loss

$$\ell_{ij} = \textsc{Bleu}(\mathbf{e}_i^*) - \textsc{Bleu}(\mathbf{e}_{ij}) \quad (3)$$

- Update $\mathbf{w}$ to the value of $\mathbf{w}'$ that minimizes:

$$\frac{1}{2}\|\mathbf{w}' - \mathbf{w}\|^2 + C \sum_{i=1}^{m} \max_{1 \le j \le n} (\ell_{ij} - \Delta\mathbf{h}_{ij} \cdot \mathbf{w}') \quad (4)$$

  where $C = 0.01$. This minimization is performed by a variant of sequential minimal optimization (Platt, 1998).

Following Chiang et al. (2008), we calculate the sentence Bleu scores in (1), (2), and (3) in the context of some previous 1-best translations. We run 20 of these learners in parallel, and when training is finished, the weight vectors from all iterations of all learners are averaged together.

Since the interface between the trainer and the decoder is fairly simple—for each sentence, the decoder sends the trainer a forest, and the trainer returns a weight update—it is easy to use this algorithm with a variety of CKY-based decoders: here, we are using it in conjunction with both the Hiero decoder and our syntax-based decoder.

4 Features

In this section, we describe the new features introduced on top of our baseline systems.

Discount features Both of our systems calculate several features based on observed counts of rules in the training data. Though the syntax-based system uses Good-Turing discounting when computing the $P(e, c)$ feature, we find, as noted above, that it uses quite a few one-count rules, suggesting that their probabilities have been overestimated. We can directly attack this problem by adding features $\mathbf{count}_i$ that reward or punish rules seen i times, or features $\mathbf{count}_{[i,j]}$ for rules seen between i and j times.

4.1 Target-side features

String-to-tree MT offers some unique levers to pull, in terms of target-side features. Because the system outputs English trees, we can analyze output trees on the tuning set and design new features to encourage the decoder to produce more grammatical trees.

Rule overlap features While individual rules observed in decoder output are often quite reasonable, two adjacent rules can create problems. For example, a rule that has a variable of type IN (preposition) needs another rule rooted with IN to fill the position. If the second rule supplies the wrong preposition, a bad translation results. The IN node here is an *overlap point* between rules. Considering that certain nonterminal symbols may be more reliable overlap points than others, we create a binary feature for each nonterminal. A rule like:

$$\text{IN(at)} \leftrightarrow \text{zai}$$

will have feature **rule-root-IN** set to 1 and all other rule-root features set to 0. Our rule root features range over the original (non-split) nonterminal set; we have 105 in total. Even though the rule root features are locally attached to individual rules—and therefore cause no additional problems for the decoder search—they are aimed at problematic rule/rule interactions.

Bad single-level rewrites Sometimes the decoder uses questionable rules, for example:

$$\text{PP}(x_0\text{:VBN } x_1\text{:NP-C}) \leftrightarrow x_0\ x_1$$

This rule is learned from 62 cases in our training data, where the VBN is almost always the word *given*. However, the decoder misuses this rule with other VBNs. So we can add a feature that penalizes any rule in which a PP dominates a VBN and NP-C. The feature class **bad-rewrite** comprises penalties for the following configurations based on our analysis of the tuning set:

$$\begin{aligned}\text{PP} &\rightarrow \text{VBN NP-C} \\ \text{PP-BAR} &\rightarrow \text{NP-C IN} \\ \text{VP} &\rightarrow \text{NP-C PP} \\ \text{CONJP} &\rightarrow \text{RB IN}\end{aligned}$$

Node count features It is possible that the decoder creates English trees with too many or too few nodes of a particular syntactic category. For example, there may be an tendency to generate too many determiners or past-tense verbs. We therefore add a count feature for each of the 109 (non-split) English nonterminal symbols. For a rule like

NPB(NNP(us) NNP(president) x_0:NNP)
$\leftrightarrow$ meiguo zongtong x_0

the feature **node-count-NPB** gets value 1, **node-count-NNP** gets value 2, and all others get 0.

Insertion features Among the rules we extract from bilingual corpora are target-language *insertion* rules, which have a word on the English side, but no words on the source Chinese side. Sample syntax-based insertion rules are:

NPB(DT(the) x_0:NN) $\leftrightarrow$ x_0
S(x_0:NP-C VP(VBZ(is) x_1:VP-C)) $\leftrightarrow$ x_0 x_1

We notice that our decoder, however, frequently fails to insert words like *is* and *are*, which often have no equivalent in the Chinese source. We also notice that *the*-insertion rules sometimes have a good effect, as in the translation "in the bloom of youth," but other times have a bad effect, as in "people seek areas of the conspiracy."

Each time the decoder uses (or fails to use) an insertion rule, it incurs some risk. There is no guarantee that the interaction of the rule probabilities and the language model provides the best way to manage this risk. We therefore provide MIRA with a feature for each of the most common English words appearing in insertion rules, e.g., **insert-the** and **insert-is**. There are 35 such features.

4.2 Source-side features

We now turn to features that make use of source-side context. Although these features capture dependencies that cross boundaries between rules, they are still local in the sense that no new states need to be added to the decoder. This is because the entire source sentence, being fixed, is always available to every feature.

Soft syntactic constraints Neither of our systems uses source-side syntactic information; hence, both could potentially benefit from soft syntactic constraints as described by Marton and Resnik (2008). In brief, these features use the output of an independent syntactic parser on the source sentence, rewarding decoder constituents that match syntactic constituents and punishing decoder constituents that cross syntactic constituents. We use separately-tunable features for each syntactic category.

Structural distortion features Both of our systems have rules with variables that generalize over possible fillers, but neither system's basic model conditions a rule application on the size of a filler, making it difficult to distinguish long-distance reorderings from short-distance reorderings. To remedy this problem, Chiang et al. (2008) introduce a structural distortion model, which we include in our experiment. Our syntax-based baseline includes the generative version of this model already.

Word context During rule extraction, we retain word alignments from the training data in the extracted rules. (If a rule is observed with more than one set of word alignments, we keep only the most frequent one.) We then define, for each triple (f, e, f_{+1}), a feature that counts the number of times that f is aligned to e and f_{+1} occurs to the right of f; and similarly for triples (f, e, f_{-1}) with f_{-1} occurring to the left of f. In order to limit the size of the model, we restrict words to be among the 100 most frequently occurring words from the training data; all other words are replaced with a token <unk>.

These features are somewhat similar to features used by Watanabe et al. (2007), but more in the spirit of features used in the word sense disambiguation model introduced by Lee and Ng (2002) and incorporated as a submodel of a translation system by Chan et al. (2007); here, we are incorporating some of its features directly into the translation model.

5 Experiments

For our experiments, we used a 260 million word Chinese/English bitext. We ran GIZA++ on the entire bitext to produce IBM Model 4 word alignments, and then the link deletion algorithm (Fossum et al., 2008) to yield better-quality alignments. For

System	Training	Features	#	Tune	Test
Hiero	MERT	baseline	11	35.4	36.1
	MIRA	syntax, distortion	56	35.9	36.9*
		syntax, distortion, discount	61	36.6	37.3**
		all source-side, discount	10990	38.4	37.6**
Syntax	MERT	baseline	25	38.6	39.5
	MIRA	baseline	25	38.5	39.8*
		overlap	132	38.7	39.9*
		node count	136	38.7	40.0**
		all target-side, discount	283	39.6	40.6**

Table 1: Adding new features with MIRA significantly improves translation accuracy. Scores are case-insensitive IBM BLEU scores. $*$ or $**$ = significantly better than MERT baseline ($p < 0.05$ or 0.01, respectively).

the syntax-based system, we ran a reimplementation of the Collins parser (Collins, 1997) on the English half of the bitext to produce parse trees, then restructured and relabeled them as described in Section 3.2. Syntax-based rule extraction was performed on a 65 million word subset of the training data. For Hiero, rules with up to two nonterminals were extracted from a 38 million word subset and phrasal rules were extracted from the remainder of the training data.

We trained three 5-gram language models: one on the English half of the bitext, used by both systems, one on one billion words of English, used by the syntax-based system, and one on two billion words of English, used by Hiero. Modified Kneser-Ney smoothing (Chen and Goodman, 1998) was applied to all language models. The language models are represented using randomized data structures similar to those of Talbot et al. (2007).

Our tuning set (2010 sentences) and test set (1994 sentences) were drawn from newswire data from the NIST 2004 and 2005 evaluations and the GALE program (with no overlap at either the segment or document level). For the source-side syntax features, we used the Berkeley parser (Petrov et al., 2006) to parse the Chinese side of both sets.

We implemented the source-side context features for Hiero and the target-side syntax features for the syntax-based system, and the discount features for both. We then ran MIRA on the tuning set with 20 parallel learners for Hiero and 73 parallel learners for the syntax-based system. We chose a stopping iteration based on the BLEU score on the tuning set, and used the averaged feature weights from all iterations of all learners to decode the test set.

Syntax-based		Hiero	
count	weight	count	weight
1	+1.28	1	+2.23
2	+0.35	2	+0.77
3–5	−0.73	3	+0.54
6–10	−0.64	4	+0.29
		5+	−0.02

Table 2: Weights learned for discount features. Negative weights indicate bonuses; positive weights indicate penalties.

The results (Table 1) show significant improvements in both systems ($p < 0.01$) over already very strong MERT baselines. Adding the source-side and discount features to Hiero yields a +1.5 BLEU improvement, and adding the target-side syntax and discount features to the syntax-based system yields a +1.1 BLEU improvement. The results also show that for Hiero, the various classes of features contributed roughly equally; for the syntax-based system, we see that two of the feature classes make small contributions but time constraints unfortunately did not permit isolated testing of all feature classes.

6 Analysis

How did the various new features improve the translation quality of our two systems? We begin by examining the discount features. For these features, we used slightly different schemes for the two systems, shown in Table 2 with their learned feature weights. We see in both cases that one-count rules are strongly penalized, as expected.

Reward		Penalty	
−0.42	a	+0.67	of
−0.13	are	+0.56	the
−0.09	at	+0.47	*comma*
−0.09	on	+0.13	*period*
−0.05	was	+0.11	in
−0.05	from	+0.08	for
−0.04	's	+0.06	to
−0.04	by	+0.05	will
−0.04	is	+0.04	and
−0.03	it	+0.02	as
−0.03	its	+0.02	have
⋮		⋮	

Table 3: Weights learned for inserting target English words with rules that lack Chinese words.

6.1 Syntax features

Table 3 shows word-insertion feature weights. The system rewards insertion of forms of *be*; examples 1–3 in Figure 1 show typical improved translations that result. Among determiners, inserting *a* is rewarded, while inserting *the* is punished. This seems to be because *the* is often part of a fixed phrase, such as *the White House*, and therefore comes naturally as part of larger phrasal rules. Inserting *the* outside these fixed phrases is a risk that the generative model is too inclined to take. We also note that the system learns to punish unmotivated insertions of commas and periods, which get into our grammar via quirks in the MT training data.

Table 4 shows weights for rule-overlap features. MIRA punishes the case where rules overlap with an IN (preposition) node. This makes sense: if a rule has a variable that can be filled by any English preposition, there is a risk that an incorrect preposition will fill it. On the other hand, splitting at a period is a safe bet, and frees the model to use rules that dig deeper into NP and VP trees when constructing a top-level S. Table 5 shows weights for generated English nonterminals: SBAR-C nodes are rewarded and commas are punished.

The combined effect of all weights is subtle. To interpret them further, it helps to look at gross changes in the system's behavior. For example, a major error in the baseline system is to move "X said" or "X asked" from the beginning of the Chinese input to the middle or end of the English trans-

Bonus		Penalty	
−0.50	*period*	+0.93	IN
−0.39	VP-C	+0.57	NNP
−0.36	VB	+0.44	NN
−0.31	SG-C	+0.41	DT
−0.30	MD	+0.34	JJ
−0.26	VBG	+0.24	*right double quote*
−0.25	ADJP	+0.20	VBZ
−0.22	-LRB-	+0.19	NP
−0.21	VP-BAR	+0.16	TO
−0.20	NPB-BAR	+0.15	ADJP-BAR
−0.16	FRAG	+0.14	PRN-BAR
−0.16	PRN	+0.14	NML
−0.15	NPB	+0.13	*comma*
−0.13	RB	+0.12	VBD
−0.12	SBAR-C	+0.12	NNPS
−0.12	VP-C-BAR	+0.12	PRP
−0.11	-RRB-	+0.11	SG
⋮		⋮	

Table 4: Weights learned for employing rules whose English sides are rooted at particular syntactic categories.

Bonus		Penalty	
−0.73	SBAR-C	+1.30	*comma*
−0.54	VBZ	+0.80	DT
−0.54	IN	+0.58	PP
−0.52	NN	+0.44	TO
−0.51	PP-C	+0.33	NNP
−0.47	*right double quote*	+0.30	NNS
−0.39	ADJP	+0.30	NML
−0.34	POS	+0.22	CD
−0.31	ADVP	+0.18	PRN
−0.30	RP	+0.16	SYM
−0.29	PRT	+0.15	ADJP-BAR
−0.27	SG-C	+0.15	NP
−0.22	S-C	+0.15	MD
−0.21	NNPS	+0.15	HYPH
−0.21	VP-BAR	+0.14	PRN-BAR
−0.20	PRP	+0.14	NP-C
−0.20	NPB-BAR	+0.11	ADJP-C
⋮		⋮	

Table 5: Weights learned for generating syntactic nodes of various types anywhere in the English translation.

lation. The error occurs with many speaking verbs, and each time, we trace it to a different rule. The problematic rules can even be non-lexical, e.g.:

$$\text{S}(x_0\text{:NP-C } x_1\text{:VP } x_2\text{:},\ x_3\text{:NP-C } x_4\text{:VP } x_5\text{:.}) \leftrightarrow x_3\ x_4\ x_2\ x_0\ x_1\ x_5$$

It is therefore difficult to come up with a straightforward feature to address the problem. However, when we apply MIRA with the features already listed, these translation errors all disappear, as demonstrated by examples 4–5 in Figure 1. Why does this happen? It turns out that in translation hypotheses that move "X said" or "X asked" away from the beginning of the sentence, more commas appear, and fewer S-C and SBAR-C nodes appear. Therefore, the new features work to discourage these hypotheses. Example 6 shows additionally that commas next to speaking verbs are now correctly deleted.

Examples 7–8 in Figure 1 show other kinds of unanticipated improvements. We do not have space for a fuller analysis, but we note that the specific effects we describe above account for only part of the overall BLEU improvement.

6.2 Word context features

In Table 6 are shown feature weights learned for the word-context features. A surprising number of the highest-weighted features have to do with translations of dates and bylines. Many of the penalties seem to discourage spurious insertion or deletion of frequent words (*for*, *'s*, *said*, parentheses, and quotes). Finally, we note that several of the features (the third- and eighth-ranked reward and twelfth-ranked penalty) shape the translation of *shuo* 'said', preferring translations with an overt complementizer *that* and without a comma. Thus these features work together to attack a frequent problem that our target-syntax features also addressed.

Figure 2 shows the performance of Hiero with all of its features on the tuning and test sets over time. The scores on the tuning set rise rapidly, and the scores on the test set also rise, but much more slowly, and there appears to be slight degradation after the 18th pass through the tuning data. This seems in line with the finding of Watanabe et al. (2007) that with on the order of 10,000 features, overfitting is possible, but we can still improve accuracy on new data.

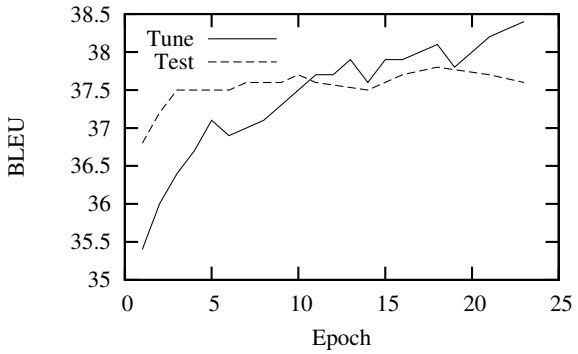

Figure 2: Using over 10,000 word-context features leads to overfitting, but its detrimental effects are modest. Scores on the tuning set were obtained from the 1-best output of the online learning algorithm, whereas scores on the test set were obtained using averaged weights.

Early stopping would have given +0.2 BLEU over the results reported in Table 1.[1]

7 Conclusion

We have described a variety of features for statistical machine translation and applied them to syntax-based and hierarchical systems. We saw that these features, discriminatively trained using MIRA, led to significant improvements, and took a closer look at the results to see how the new features qualitatively improved translation quality. We draw three conclusions from this study.

First, we have shown that these new features can improve the performance even of top-scoring MT systems. Second, these results add to a growing body of evidence that MIRA is preferable to MERT for discriminative training. When training over 10,000 features on a modest amount of data, we, like Watanabe et al. (2007), did observe overfitting, yet saw improvements on new data. Third, we have shown that syntax-based machine translation offers possibilities for features not available in other models, making syntax-based MT and MIRA an especially strong combination for future work.

[1] It was this iteration, in fact, which was used to derive the combined feature count used in the title of this paper.

1 MERT: the united states pending israeli clarification on golan settlement plan
MIRA: the united states is waiting for israeli clarification on golan settlement plan

2 MERT: . . . the average life expectancy of only 18 months , canada 's minority goverment will . . .
MIRA: . . . the average life expectancy of canada's previous minority government is only 18 months . . .

3 MERT: . . . since un inspectors expelled by north korea . . .
MIRA: . . . since un inspectors were expelled by north korea . . .

4 MERT: another thing is . . . , " he said , " obviously , the first thing we need to do . . . .
MIRA: he said : " obviously , the first thing we need to do . . . , and another thing is . . . . "

5 MERT: the actual timing . . . reopened in january , yoon said .
MIRA: yoon said the issue of the timing . . .

6 MERT: . . . us - led coalition forces , said today that the crash . . .
MIRA: . . . us - led coalition forces said today that a us military . . .

7 MERT: . . . and others will feel the danger .
MIRA: . . . and others will not feel the danger .

8 MERT: in residential or public activities within 200 meters of the region , . . .
MIRA: within 200 m of residential or public activities area , . . .

Figure 1: Improved syntax-based translations due to MIRA-trained weights.

Bonus			
	f	e	context
−1.19	<unk>	<unk>	f_{-1} = ri 'day'
−1.01	<unk>	<unk>	f_{-1} = (
−0.84	,	that	f_{-1} = shuo 'say'
−0.82	yue 'month'	<unk>	f_{+1} = <unk>
−0.78	"	"	f_{-1} = <unk>
−0.76	"	"	f_{+1} = <unk>
−0.66	<unk>	<unk>	f_{+1} = nian 'year'
−0.65	,	that	f_{+1} = <unk>
		⋮	

Penalty			
	f	e	context
+1.12	<unk>	)	f_{+1} = <unk>
+0.83	jiang 'shall'	be	f_{+1} = <unk>
+0.83	zhengfu 'government'	the	f_{-1} = <unk>
+0.73	<unk>	)	f_{-1} = <unk>
+0.73	<unk>	(	f_{+1} = <unk>
+0.72	<unk>	)	f_{-1} = ri 'day'
+0.70	<unk>	(	f_{-1} = ri 'day'
+0.69	<unk>	(	f_{-1} = <unk>
+0.66	<unk>	for	f_{-1} = <unk>
+0.66	<unk>	's	f_{-1} = ,
+0.65	<unk>	said	f_{-1} = <unk>
+0.60	,	,	f_{-1} = shuo 'say'
		⋮	

Table 6: Weights learned for word-context features, which fire when English word e is generated aligned to Chinese word f, with Chinese word f_{-1} to the left or f_{+1} to the right. Glosses for Chinese words are not part of features.

References

Phil Blunsom, Trevor Cohn, and Miles Osborne. 2008. A discriminative latent variable model for statistical machine translation. In *Proc. ACL-08: HLT*.

Peter F. Brown, Stephen A. Della Pietra, Vincent Della J. Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–312.

Yee Seng Chan, Hwee Tou Ng, and David Chiang. 2007. Word sense disambiguation improves statistical machine translation. In *Proc. ACL 2007*.

Stanley F. Chen and Joshua T. Goodman. 1998. An empirical study of smoothing techniques for language modeling. Technical Report TR-10-98, Computer Science Group, Harvard University.

David Chiang, Yuval Marton, and Philip Resnik. 2008. Online large-margin training of syntactic and structural translation features. In *Proc. EMNLP 2008*.

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

David Chiang. 2007. Hierarchical phrase-based translation. *Computational Linguistics*, 33(2).

Michael Collins. 1997. Three generative, lexicalized models for statistical parsing. In *Proc. ACL 1997*.

Koby Crammer, Ofer Dekel, Joseph Keshet, Shai Shalev-Shwartz, and Yoram Singer. 2006. Online passive-aggressive algorithms. *Journal of Machine Learning Research*, 7:551–585.

Steve DeNeefe, Kevin Knight, Wei Wang, and Daniel Marcu. 2007. What can syntax-based MT learn from phrase-based MT? In *Proc. EMNLP-CoNLL-2007*.

Victoria Fossum, Kevin Knight, and Steven Abney. 2008. Using syntax to improve word alignment for syntax-based statistical machine translation. In *Proc. Third Workshop on Statistical Machine Translation*.

Michel Galley, Mark Hopkins, Kevin Knight, and Daniel Marcu. 2004. What's in a translation rule? In *Proc. HLT-NAACL 2004*, Boston, Massachusetts.

Michel Galley, Jonathan Graehl, Kevin Knight, Daniel Marcu, Steve DeNeefe, Wei Wang, and Ignacio Thayer. 2006. Scalable inference and training of context-rich syntactic models. In *Proc. ACL 2006*.

Liang Huang. 2008. Forest reranking: Discriminative parsing with non-local features. In *Proc. ACL 2008*.

Kevin Knight and Jonathan Graehl. 2005. An overview of probabilistic tree transducers for natural language processing. In *Proceedings of the Sixth International Conference on Intelligent Text Processing and Computational Linguistics (CICLing)*.

Yoong Keok Lee and Hwee Tou Ng. 2002. An empirical evaluation of knowledge sources and learning algorithms for word sense disambiguation. In *Proc. EMNLP 2002*, pages 41–48.

Percy Liang, Alexandre Bouchard-Côté, Dan Klein, and Ben Taskar. 2006. An end-to-end discriminative approach to machine translation. In *Proc. COLING-ACL 2006*.

Wolfgang Macherey, Franz Josef Och, Ignacio Thayer, and Jakob Uskoreit. 2008. Lattice-based minimum error rate training for statistical machine translation. In *Proc. EMNLP 2008*.

Yuval Marton and Philip Resnik. 2008. Soft syntactic constraints for hierarchical phrased-based translation. In *Proc. ACL-08: HLT*.

Franz Josef Och and Hermann Ney. 2002. Discriminative training and maximum entropy models for statistical machine translation. In *Proc. ACL 2002*.

Franz Josef Och, Daniel Gildea, Sanjeev Khudanpur, Anoop Sarkar, Kenji Yamada, Alex Fraser, Shankar Kumar, Libin Shen, David Smith, Katherine Eng, Viren Jain, Zhen Jin, and Dragomir Radev. 2004. A smorgasbord of features for statistical machine translation. In *Proc. HLT-NAACL 2004*, pages 161–168.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proc. ACL 2003*.

Slav Petrov, Leon Barrett, Romain Thibaux, and Dan Klein. 2006. Learning accurate, compact, and interpretable tree annotation. In *Proc. ACL 2006*.

John C. Platt. 1998. Fast training of support vector machines using sequential minimal optimization. In B. Schölkopf, C. J. C. Burges, and A. J. Smola, editors, *Advances in Kernel Methods: Support Vector Learning*, pages 195–208. MIT Press.

David Talbot and Miles Osborne. 2007. Randomised language modelling for statistical machine translation. In *Proc. ACL 2007*, pages 512–519.

Christoph Tillmann and Tong Zhang. 2006. A discriminative global training algorithm for statistical MT. In *Proc. COLING-ACL 2006*.

Joseph Turian, Benjamin Wellington, and I. Dan Melamed. 2007. Scalable discriminative learning for natural language parsing and translation. In *Proc. NIPS 2006*.

Wei Wang, Kevin Knight, and Daniel Marcu. 2007. Binarizing syntax trees to improve syntax-based machine translation accuracy. In *Proc. EMNLP-CoNLL 2007*.

Taro Watanabe, Jun Suzuki, Hajime Tsukuda, and Hideki Isozaki. 2007. Online large-margin training for statistical machine translation. In *Proc. EMNLP 2007*.

Dekai Wu. 1997. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Computational Linguistics*, 23:377–404.

Kenji Yamada and Kevin Knight. 2002. A decoder for syntax-based statistical MT. In *Proc. ACL 2002*.

Hao Zhang, Liang Huang, Daniel Gildea, and Kevin Knight. 2006. Synchronous binarization for machine translation. In *Proc. HLT-NAACL 2006*.