Syntactic Re-Alignment Models for Machine Translation

Abstract

We present a method for improving word alignment for machine translation that employs a model of alignment closer to the model of translation than is commonly done. This leads to better extraction of lingusitic patterns for a syntactic machine translation system and in turn leads to improved BLEU scores on translation experiments in Chinese and Arabic.

1 Methods of statistical MT

Roughly speaking, there are two paths commonly taken in statistical machine translation (Figure 1). The idealistic path uses an unsupervised learning algorithm such as EM (Demptser et al., 1977) to learn parameters for some proposed translation model from a bitext training corpus, and then directly translates using the weighted model. Some examples of the idealistic approach are the direct IBM word model (Berger et al., 1994; Germann et al., 2001), the phrase-based approach of Marcu and Wong (2002), and the syntax approaches of Wu (1996) and Yamada and Knight (2001). Idealistic approaches are conceptually simple and thus easy to relate to observed phenomena. As more parameters are added to the model definition the idealistic approach has not scaled well, for it is increasingly difficult to incorporate large amounts of training data efficiently over an increasingly large search space. Additionally, the EM procedure has a bias to memorizing its training data and thus learning is not robust when the input units have varying explanatory powers, such as variable-size phrases or variable-height trees.

The realistic path also learns a model of translation, but uses that model only to obtain Viterbi word-for-word alignments for the training corpus. The bitext and corresponding alignments are then used as input to a pattern extraction algorithm, which yields a set of patterns or rules for a second translation model (which often has a wider parameter space than that used to obtain the word-for-word alignments). Weights for the second model are then set, typically by counting and smoothing, and this weighted model is used for translation. Realistic approaches scale to large data sets and have yielded better BLEU performance than their counterparts, but there is a disconnect between the first model (hereafter, the *alignment model*) and the second (*translation model*). Examples of realistic systems are the phrase-based ATS system of Och and Ney (2004), the phrasal-syntax hybrid system Hiero (Chiang, 2005), and the GHKM syntax system (Galley et al., 2004; Galley et al., 2006). For an alignment model, most of these use the Aachen HMM approach (Vogel et al., 1996), the implementation of IBM Model 4 in GIZA++ (Och and Ney, 2000) or, more recently, the semi-supervised EMD algorithm (Fraser and Marcu, 2006).

The two-model approach of the realistic path has undeniable empirical qualities and scales to large data sets, but new research tends to focus on development of higher order translation models that are informed only by low-order alignments. We would like to add the analytic power gained from modern translation models to the underlying alignment

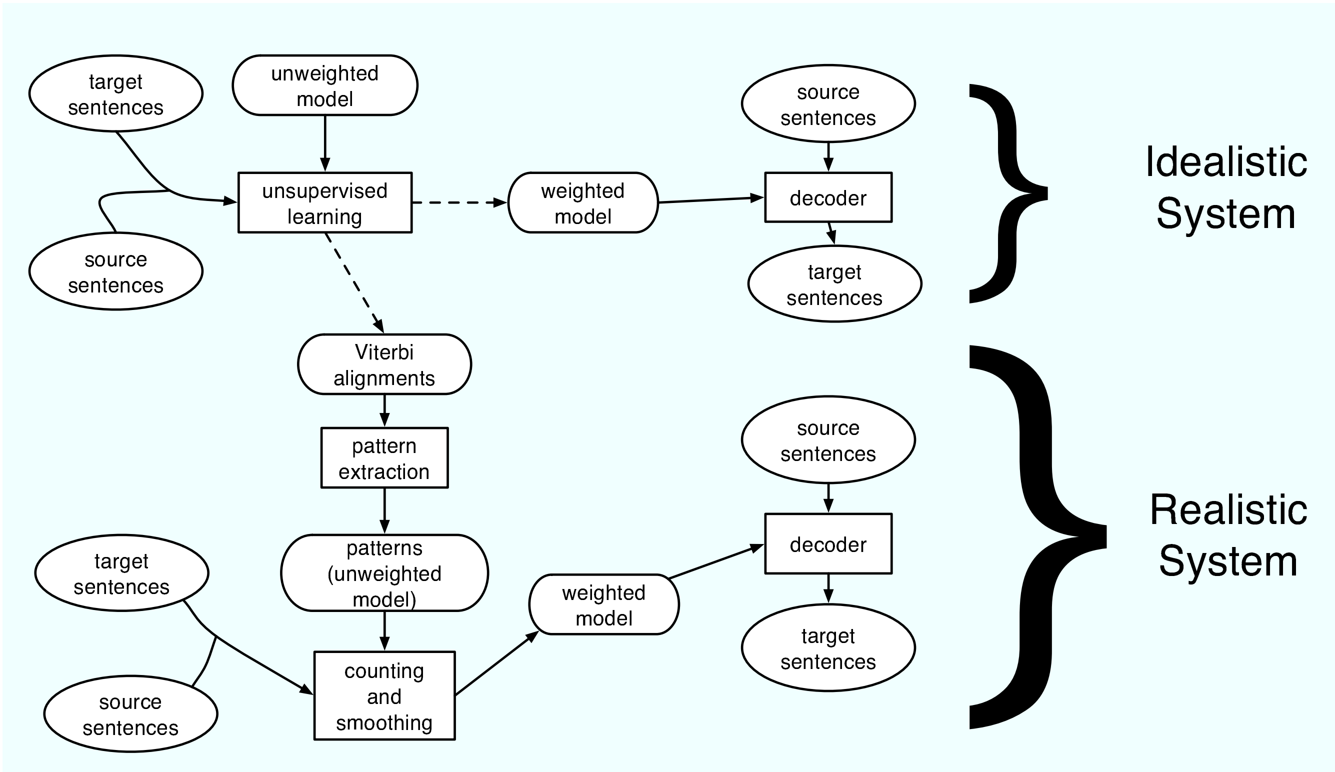

Figure 1: General approach to idealistic and realistic statistical MT systems

model without sacrificing the efficiency and empirical gains of the two-model approach. By adding the same syntactic information used in our translation model to our alignment model we may improve alignment quality in a way that improves rule quality and, in turn, system quality. In the remainder of this work we demonstrate how a touch of idealism can improve an existing realistic syntax-based translation system.

2 Multi-level syntactic rules for syntax MT

Galley et al. (2004) and Galley et al. (2006) describe a syntactic translation model that relates English trees to foreign strings. The model describes joint production of a (tree, string) pair via a nondeterministic selection of weighted rules. Each rule has an English tree fragment with variables and a corresponding foreign string fragment with the same variables. A series of rules forms an explanation (or *derivation*) of the complete pair.

As an example, consider the parsed English and corresponding Chinese at the top of Figure 2. The three columns at the bottom of the figure are examples of rule sequences that can explain this pair; there are many other possibilities. Note how rules specify rotation (e.g. R10, R5), direct translation (R12, R8), insertion and deletion (R11, R1), and tree traversal (R7, R15). Note too that the rules explain variable-size fragments (e.g. R6 vs. R14) and thus the possible *derivation trees* of rules that explain a sentence pair have varying sizes. The smallest such derivation tree has a single large rule (which does not appear in Figure 2; we leave the description of such a rule as an exercise for the reader). A string-to-tree decoder constructs a *derivation forest* of derivation trees whose rule right sides, taken together, explain a candidate source sentence. It then outputs the English tree corresponding to the highest-scoring derivation in the forest.

3 Introducing syntax into the alignment model

Now that we have described our syntax rule framework we lay the ground for a syntactically motivated

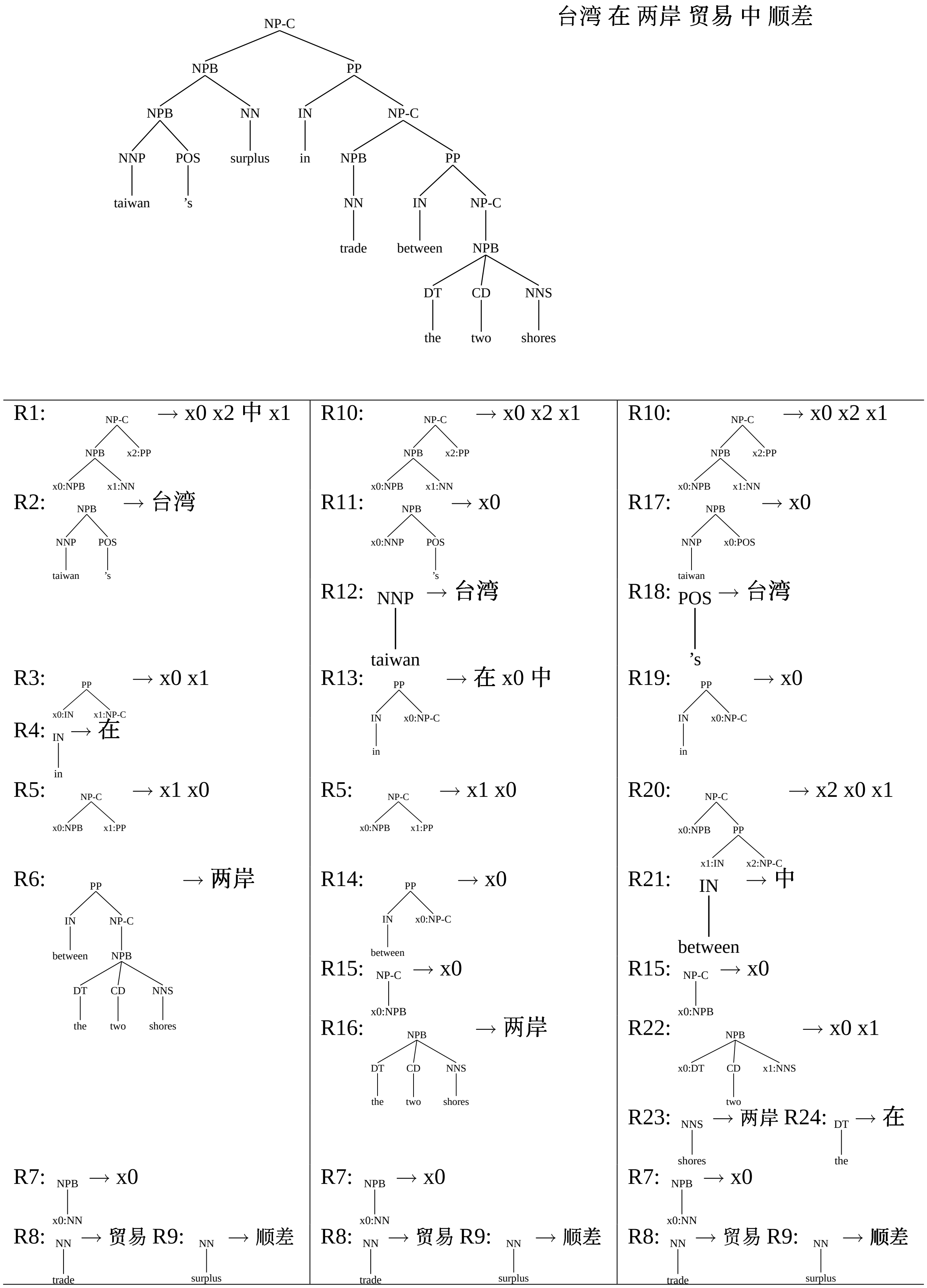

Figure 2: A (English tree, Chinese string) pair and three different sets of multilevel tree-to-string rules that can explain it; the first set is obtained from EMD alignments, the second from this paper’s re-alignment procedure, and the third is a viable, if poor, alternative that is not learned.

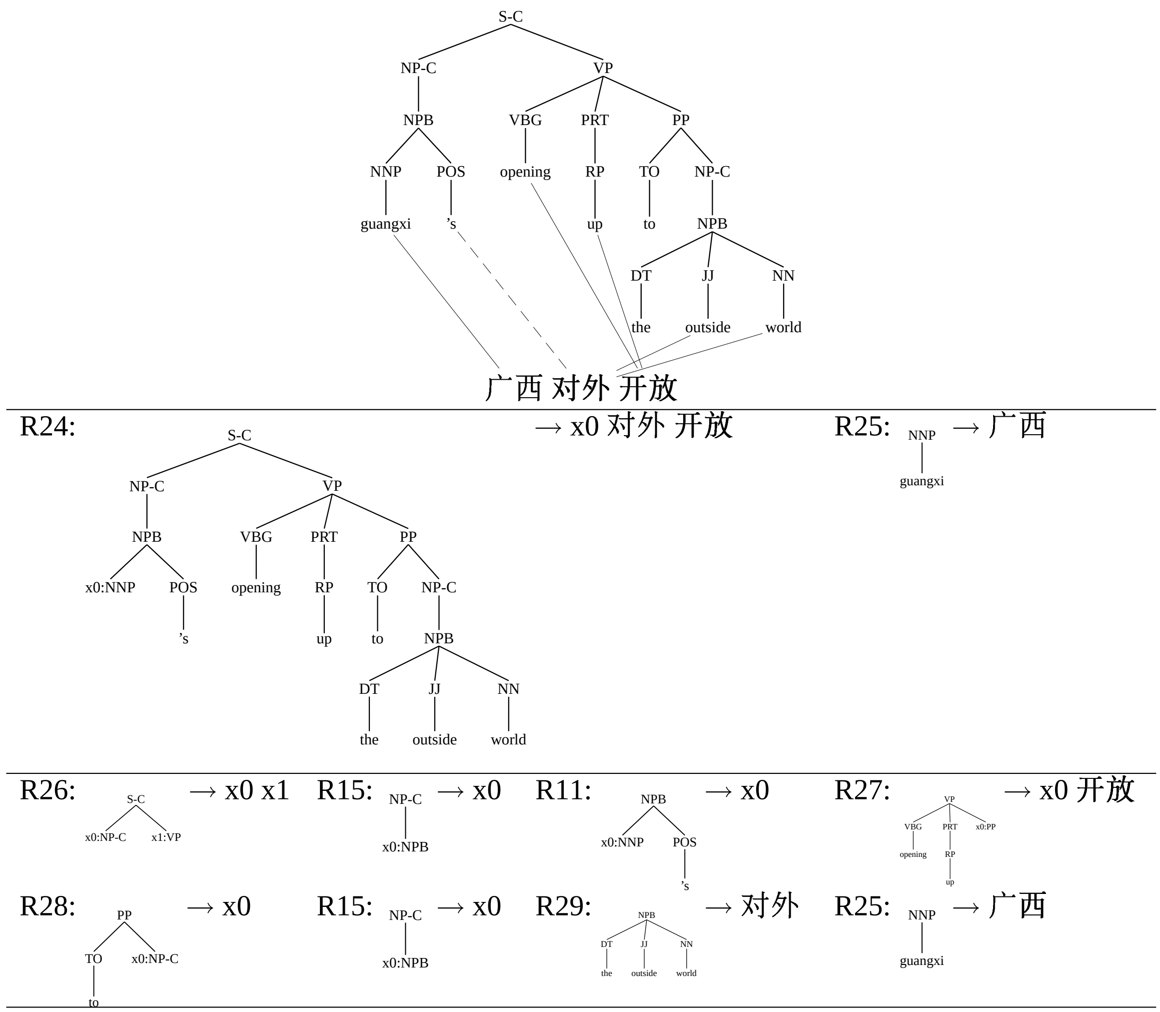

Figure 3: The impact of a bad alignment on rule extraction. The alignment link indicated by the dotted line in the example leads to the rule set in the second row. The re-alignment procedure described in Section 3.2 learns to prefer the rule set at bottom.

alignment model. We begin by reviewing an alignment model commonly seen in realistic MT systems and compare it to a syntactically-aware alignment model.

3.1 The IBM traditional alignment model

IBM Model 4 (Brown et al., 1993) learns a set of 4 probability tables to compute $p(f|e)$ given a foreign sentence f and its target translation e via the following (greatly simplified) generative story:

1. A fertility for each word e_i in e is chosen with probability $p_{fert}(e_i)$.

2. A null word is inserted next to each fertility-expanded word with probability p_{null}.

3. Each token e_i in the fertility-expanded word and null string is translated into some foreign word f_i in f with probability $p_{trans}(f_i|e_i)$.

4. The position of each foreign word f_i that was translated from e_i is changed by Δ (which may be positive, negative, or zero) with probability $p_{distortion}(\Delta|\mathcal{A}(e_i), \mathcal{B}(f_i))$, where $\mathcal{A}$ and $\mathcal{B}$ are functions over the source and target vocabularies, respectively.

Brown et al. (1993) describes an EM learning procedure for estimating values for the four tables in the generative story. However, searching the space of all possible alignments is intractable for EM, so in practice the procedure is bootstrapped by models with narrower search space such as IBM Model 1 (Brown et al., 1993) or Aachen HMM (Vogel et al., 1996).

3.2 A syntax re-alignment model

Now let us contrast this commonly used model for obtaining alignments with a syntactically motivated alternative. We recall the rules described in Section 2. Our model learns a single probability table to compute $p(etree, f)$ given a foreign sentence f and a parsed target translation $etree$. In the following generative story we assume a starting variable v.

1. Choose a rule r to replace v, with probability $p_{rule}(r|v)$.
2. For each variable $v_i$in the partially completed (tree, string) pair, continue to choose rules r_i with probability $p_{rule}(r_i|v_i)$ to replace these variables until there are no variables remaining.

In Section 5.1 we discuss an EM learning procedure for estimating these rule probabilities.

As in the IBM approach, we must mitigate tractability by limiting the parameters of the model, which are potentially much wider than in the word-to-word case. We would like to learn all possible rules that explain the training data, but this implies a rule relating each possible tree fragment to each possible string fragmentation, which is infeasible. We follow the approach of bootstrapping from a model with a narrower parameter space as is done in, e.g. Och and Ney (2000) and Fraser and Marcu (2006).

To reduce the model space we employ the rule acquisition technique of Galley et al. (2004), which obtains rules given a (tree, string) pair as well as an initial alignment between them. We are agnostic about the source of this bootstrapped alignment and in Section 5 present results based on several different bootstrapped alignment qualities. Since we require an initial set of alignments, which we obtain from a word-for-word alignment procedure such as GIZA++ or EMD, we are not aligning input data, but rather *re-aligning* it with a syntax model.

		SENTENCE PAIRS	
	DESCRIPTION	CHINESE	ARABIC
TUNE	NIST 2002 short	925	696
TEST	NIST 2003	919	663

Figure 4: Tuning and testing data sets for the MT system described in Section 5.2.

4 The appeal of a syntax alignment model

Consider the example of Figure 2 again. The leftmost derivation is that obtained from the bootstrapped alignment set. This derivation is decent but there are some poorly motivated rules, from a linguistic standpoint. The Chinese word 两岸 roughly means “the two shores” in this context, but the rule R6 learned from the alignment incorrectly includes “between”. However, some other sentence in the training corpus has the correct alignment which yields rule R16 and rules R13 and R14 (learned from yet other sentences in the training corpus) handle the fairly complicated 在 ... 中 structure (which roughly translates to “in between”), thus allowing the middle derivation.

EM distributes rule probabilities in such a way as to maximize the probability of the training corpus. It thus prefers to use one rule many times instead of several different rules for the same situation over several sentences, if possible. R6 is a possible rule in 46 of the 329,031 sentence pairs in the training corpus, while R16 is a possible rule in 100 sentence pairs. The syntactic components of these rules inform their usability, and thus well-formed rules are favored over ill-formed rules. This allows EM to learn to disfavor alignments with long-distance crossings. The top row of Figure 3 contains an example of a long-distance crossing learned by the bootstrapped alignment model. The rules extracted from this alignment, seen on the second row of the figure, are poor. A set of commonly seen rules learned from other training sentences provide a more likely explanation of the data, and the alignment learned by this superior derivation omits the spurious link.

5 Experiments

In this section, we describe the implementation of our semi-idealistic model and our means of evaluat-

BOOTSTRAP GIZA CORPUS		RE-ALIGNMENT EXPERIMENT			
ENGLISH WORDS	CHINESE WORDS	TYPE	RULES	TUNE	TEST
9,864,294	7,520,779	baseline	19,138,252	39.08	37.77
		initial	18,698,549	39.49	38.39
		adjusted	26,053,341	**39.76**	**38.69**

Figure 5: A comparison of Chinese BLEU performance between the GIZA baseline (no re-alignment), re-alignment as proposed in Section 3.2, and re-alignment as modified in Section 5.4

ing the quality of the alignments in a machine translation task.

5.1 The re-alignment setup

We began with a training corpus of Chinese-English and Arabic-English bitexts, the English side parsed by a reimplementation of the standard Collins model (Bikel, 2004). In order to acquire a syntactic rule set, we also need a bootstrapped alignment of each training sentence. We used an implementation of the GHKM algorithm (Galley et al., 2004) to obtain a rule set for each bootstrapped alignment.

Now we need an EM algorithm for learning the parameters of the rule set that maximize $\prod_{corpus} p(tree, string)$. Such an algorithm is presented by Graehl and Knight (2004). The algorithm consists of two components: DERIV, which is a procedure for constructing a packed forest of derivation trees of rules that explain a (tree, string) bitext corpus given that corpus and a rule set, and TRAIN, which is an iterative parameter-setting procedure.

We initially attempted to use the top-down DERIV algorithm of Graehl and Knight (2004), but as the constraints of the derivation forests are largely lexical, too much time was spent on exploring dead-ends. Instead we built derivation forests from a forced-decoding algorithm. The standard CKY-style decoder used for machine translation was configured to limit its search to the training pair, and thus we were able to build derivation forests using a bottom-up approach, which encounters leaf lexical constraints immediately and thus has a narrower search space and yields faster performance than the top-down algorithm. Building derivation forests took around 400 hours of cumulative machine time (4-processor machines) for Chinese. The actual running of EM iterations (which directly implements the TRAIN algorithm of Graehl and Knight (2004)) took about 10 minutes, after which the Viterbi derivation tree was directly recoverable. The Viterbi derivation tree tells us which English words produced which Chinese words, so we can extract a word-to-word alignment from it.

5.2 The MT system setup

A truly idealistic MT system would then directly apply the rule weight parameters learned via EM to a machine translation task. As mentioned in Section 1, we maintain the two-model, or realistic approach. Below we briefly describe the translation model, focusing on comparison with the previously described alignment model. For a more complete description of the translation model we refer the reader to (Galley et al., 2006).

Although in principle the re-alignment model and translation model learn parameter weights over the same rule space, in practice we limit the rules used for re-alignment to the set of smallest rules that explain the training corpus and are consistent with the bootstrap alignments. This is a compromise made to reduce the search space for EM. The translation model learns multiple derivations of rules consistent with the re-alignments for each sentence, and learns weights for these by counting and smoothing. A dozen other features are also added to the rules. We obtain lambda weights for the combinations of the features by performing minimum error rate training (Och, 2003) on held-out data. We then use a CKY decoder to translate unseen test data using the rules and tuned lambdas. Figure 4 summarizes the data used in tuning and testing.

5.3 Initial results

An initial re-alignment experiment showed a reasonable rise in BLEU scores from the baseline (Figure 5), but closer inspection of the rules favored by EM implied we could do even better. EM has a tendency

BOOTSTRAP GIZA CORPUS		RE-ALIGNMENT EXPERIMENT			
ENGLISH WORDS	CHINESE WORDS	TYPE	RULES	TUNE	TEST
9,864,294	7,520,779	baseline	19,138,252	39.08	37.77
		re-alignment	26,053,341	**39.76**	**38.69**
221,835,870	203,181,379	baseline	23,386,535	39.51	38.93
		re-alignment	33,374,646	**40.17**	**39.96**

(a) Chinese re-alignment corpus has 9,864,294 English and 7,520,779 Chinese words

BOOTSTRAP GIZA CORPUS		RE-ALIGNMENT EXPERIMENT			
ENGLISH WORDS	ARABIC WORDS	TYPE	RULES	TUNE	TEST
4,067,454	3,147,420	baseline	2,333,839	**47.92**	47.33
		re-alignment	2,474,737	47.87	**47.89**
168,255,347	147,165,003	baseline	3,245,499	49.72	49.60
		re-alignment	3,600,915	49.73	**49.99**

(b) Arabic re-alignment corpus has 4,067,454 English and 3,147,420 Arabic words

Figure 6: Machine Translation experimental results. BLEU is case-insensitive BLEU4 evaluated on untokenized references.

to favor few large rules over many small rules, even when the small rules are more useful. Referring to the rules in Figure 2, note that possible derivations for (taiwan 's, 台湾)[1] are R2, R11-R12, and R17-R18. Clearly the third derivation is not desirable, and we do not discuss it further. Between the first two derivations, R11-R12 is preferred over R2, as the conditioning for possessive insertion is not related to the specific Chinese word being inserted. Of the 1902 sentences in the training corpus where this pair is seen, the bootstrapped alignments yield the R2 derivation 1649 times and the R11-R12 derivation 0 times. The result after re-alignment is little changed; the new alignments yield the R2 derivation 1613 times and again never choose R11-R12. The rules in the second derivation themselves are not rarely seen – R11 is in 13,311 forests *other* than those where R2 is seen, and R12 is in 2500 additional forests. EM gives R11 a probability of $e^{-7.72}$ – more likely than 98.7% of rules, and R12 a probability of $e^{-2.96}$. But R2 receives a probability of $e^{-6.32}$ and is thus preferred over the R11-R12 derivation, which has a combined probability of $e^{-10.68}$.

5.4 Making EM fair

The preference of shorter derivations of large rules over longer derivations of small rules is due to a general tendency for EM to prefer derivations with few atoms. This has been noted before, e.g. by Marcu and Wong (2002). As the probability of a derivation is determined by the product of its atom probabilities, longer derivations with more probabilities to multiply have an inherent disadvantage against shorter derivaions, all else being equal. EM is an iterative procedure and thus such a bias can lead the procedure to converge with artificially raised probabilities for short derivations and the large rules that comprise them. The relatively rare applicability of large rules (and thus lower observed partial counts) does not overcome the inherent advantage of large coverage. To combat this, we introduce size terms into our generative story, ensuring that all competing derivations contain the same number of atoms:

1. Choose a rule size s with probability $p_{size}(s)^{s-1}$.
2. Choose a rule r (of size s) to replace the start symbol with probability $p_{rule}(r)$.
3. For each variable in the partially completed (tree, string) pair, continue to choose sizes followed by rules, recursively to replace these variables until there are no variables remaining.

This generative story changes the derivation comparison from R2 vs R11-R12 to S2-R2 vs R11-R12, where S2 is the atom that represents the choice of

[1] The Chinese gloss is simply "taiwan".

LANGUAGE PAIR	TYPE	RULES	TUNE	TEST
CHINESE-ENGLISH	baseline	55,781,061	**42.79**	**41.83**
	EMD re-align	69,318,930	42.67	41.82
ARABIC-ENGLISH	baseline	8,487,656	**52.33**	52.25
	EMD re-align	11,498,150	52.3	**52.71**

Figure 7: Re-alignment performance with semi-supervised EMD bootstrap alignments

size 2 (the size of a rule in this context is the number of non-leaf and non-root nodes in its tree fragment). Note that the variable number of inclusions implied by the exponent in the generative story above ensures that all derivations have the same size. For example, a derivation with one size-3 rule, a derivation with one size-2 and one size-1 rules, and a derivation with three size-1 rules would each have three atoms. With this revised model that allows for fair comparison of derivations, the R11-R12 derivation is chosen 1636 times, and S2-R2 is not chosen. R2 does, however, appear in the translation model, as the expanded rule extraction described in Section 5.2 creates R2 by joining R11 and R12.

The probability of size atoms, like the probability of rule atoms, is decided by EM. However, note that the generative story tends to encourage smaller sizes by virtue of the exponent. This does not, however, simply ensure the largest number of rules per derivation is used in all cases. Ill-fitting and poorly-motivated rules such as R22, R23, and R24 in Figure 2 are not preferred over R16, even though they are smaller. However, R16 is preferred over R6, as the former is a useful rule. The adjusted generative story leads to an improvement in BLEU score, as can be seen in the last row of Figure 5.

5.5 Results

We performed primary experiments on two different boostrapped setups in two languages: the initial experiment uses the same data set for the GIZA++ initial alignment as is used in the re-alignment, while an experiment on better quality bootstrapped alignments uses a much larger data set. For each bootstrapping in each language we compared the baseline of using these alignments directly in an MT system with the experiment of using the alignments obtained from the re-alignment procedure described in Section 5.1. For each experiment we report: the number of rules extracted by the expanded GHKM algorithm of Galley et al. (2006) for the translation model, converged BLEU scores on the tuning set, and finally BLEU performance on the held-out test set. Data set specifics for the GIZA++ bootstrapping and BLEU results are summarized in Figure 6.

5.6 Discussion

The results presented demonstrate we are able to improve on unsupervised GIZA++ alignments by about 1 BLEU point for Chinese and around 0.4 BLEU point for Arabic using an additional unsupervised algorithm that requires no human aligned data. If human-aligned data is available, the EMD algorithm provides higher baseline alignments than GIZA++ that have led to better MT performance (Fraser and Marcu, 2006). As a further experiment we repeated the experimental conditions from Figure 6, this time bootstrapped with the semi-supervised EMD method, which uses the larger bootstrap GIZA corpora described in Figure 6 and an additional 64,469/48,650 words of hand-aligned English-Chinese and 43,782/31,457 words of hand-aligned English-Arabic. The results of this advanced experiment are in Figure 7. We show a 0.46 gain in BLEU for Arabic, but no movement for Chinese. We believe increasing the size of the re-alignment corpora will increase BLEU gains in this experimental condition, but leave those results for future work.

We can see from the results presented that the impact of the syntax-aware re-alignment procedure of Section 3.2, coupled with the addition of size parameters to the generative story from Section 5.4 serves to remove links from the bootstrapped alignments that cause less useful rules to be extracted, and thus increase the overall quality of the rules, and hence the system performance. We thus see the benefit to including syntax in an alignment model, bringing the two models of the realistic machine translation path somewhat closer together.

References

Adam Berger, Peter Brown, Stephen Della Pietra, Vincent Della Pietra, John Gillett, John Lafferty, Robert Mercer, Harry Printz, and Luboš Ureš. 1994. The candide system for machine translation. In *Proceedings of the ARPA Workshop on Human Language Technology*, pages 157–162, Plainsboro, New Jersey, USA, March 8–11.

Daniel Bikel. 2004. Intricacies of collins' parsing model. *Computational Linguistics*, 30(4):479–511.

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

David Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pages 263–270, Ann Arbor, Michigan, USA, June 25–30.

Arthur P. Demptser, Nan M. Laird, and Donald B. Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38.

Alexander Fraser and Daniel Marcu. 2006. Semi-supervised training for statistical word alignment. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 769–776, Sydney, Australia, July 17–21.

Michel Galley, Mark Hopkins, Kevin Knight, and Daniel Marcu. 2004. What's in a translation rule? In *HLT-NAACL 2004: Main Proceedings*, pages 273–280, Boston, Massachusetts, USA, May 2–7.

Michel Galley, Jonathan Graehl, Kevin Knight, Daniel Marcu, Steven DeNeefe, Wei Wang, and Ignacio Thayer. 2006. Scalable inference and training of context-rich syntactic models. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 961–968, Sydney, Australia, July 17–21.

Ulrich Germann, Michael Jahr, Kevin Knight, Daniel Marcu, and Kenji Yamada. 2001. Fast decoding and optimal decoding for machine translation. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*, pages 228–235, Toulouse, France, July 9–11.

Jonathan Graehl and Kevin Knight. 2004. Training tree transducers. In *HLT-NAACL 2004: Main Proceedings*, pages 105–112, Boston, Massachusetts, USA, May 2–7.

Daniel Marcu and William Wong. 2002. A phrase-based, joint probability model for statistical machine translation. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing*, pages 133–139, Philadelphia, Pennsylvania, USA, July 6–7.

Franz Och and Hermann Ney. 2000. Improved statistical alignment models. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, pages 440–447, Hong Kong, October 1–8.

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

Franz Och. 2003. Minimum error rate training for statistical machine translation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 160–167, Sapporo, Japan, July 8–10.

Stephan Vogel, Hermann Ney, and Christoph Tillmann. 1996. HMM-based word alignment in statistical translation. In *COLING96: Proceedings of the 16th International Conference on Computational Linguistics*, pages 836–841, Copenhagen, Denmark, August 5–9.

Dekai Wu. 1996. A polynomial-time algorithm for statistical machine translation. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, pages 152–158, Santa Cruz, California, USA, June 24–27.

Kenji Yamada and Kevin Knight. 2001. A syntax-based statistical translation model. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*, pages 523–530, Toulouse, France, July 9–11.