Feature-Rich Language-Independent Syntax-Based Alignment for Statistical Machine Translation

Jason Riesa[1] **Ann Irvine**[2] **Daniel Marcu**[1]

[1]Information Sciences Institute
University of Southern California
Marina del Rey, CA 90292
{riesa, marcu}@isi.edu

[2]Department of Computer Science
Johns Hopkins University
Baltimore, MD 21218
anni@jhu.edu

Abstract

We present an accurate word alignment algorithm that heavily exploits source and target-language syntax. Using a discriminative framework and an efficient bottom-up search algorithm, we train a model of hundreds of thousands of syntactic features. Our new model (1) helps us to very accurately model syntactic transformations between languages; (2) is language-independent; and (3) with automatic feature extraction, assists system developers in obtaining good word-alignment performance off-the-shelf when tackling new language pairs. We analyze the impact of our features, describe inference under the model, and demonstrate significant alignment and translation quality improvements over already-powerful baselines trained on very large corpora. We observe translation quality improvements corresponding to 1.0 and 1.3 BLEU for Arabic-English and Chinese-English, respectively.

1 Introduction

In recent years, several state-of-the-art statistical machine translation (MT) systems have incorporated both source and target syntax into the grammars that they generate and use to translate. While some tree-to-tree systems parse source and target sentences separately (Galley et al., 2006; Zollman and Venugopal, 2006; Huang and Mi, 2010), others project syntactic parses across word alignments (Li et al., 2009). In both approaches, as in largely all statistical MT, the quality of the alignments used to generate the rules of the grammar are critical to the success of the system. However, to date, most word alignment systems have not considered the same degree of syntactic information that MT systems have.

Extending unsupervised models, like the IBM models (Brown et al., 1993), generally requires changing the entire generative story. The additional complexity would likely make training such models quite expensive. Already, with ubiquitous tools like GIZA++ (Och and Ney, 2003), training accurate models on large corpora takes upwards of 5 days.

Recent work in discriminative alignment has focused on incorporating features that are unavailable or difficult to incorporate within other models, e.g. (Moore, 2005; Ittycheriah and Roukos, 2005; Liu et al., 2005; Taskar et al., 2005b; Blunsom and Cohn, 2006; Lacoste-Julien et al., 2006; Moore et al., 2006). Even more recently, motivated by the rise of syntax-based translation models, others have sought to inform alignment decisions with syntactic information (Fraser and Marcu, 2007; DeNero and Klein, 2007; May and Knight, 2007; Fossum et al., 2008; Haghighi et al., 2009; Burkett et al., 2010; Pauls and Klein, 2010; Riesa and Marcu, 2010).

Motivated by the wide modeling gap that still remains between syntax-based translation and word-alignment models, in this paper we expand on previous work in discriminative alignment, and move forward in three key areas:

1. We *heavily exploit both source and target syntax* in ways that most models can not. In addition, during training we extract and learn hundreds of thousands of features automatically, learning both the structure and parameters for the model at the same time.

2. Our model and inference support *arbitrary features*, and easily scale to millions of features.

3. Having strengthened the synchronicity between

alignment and syntax-based translation models, we *advance state-of-the-art performance* in terms of both alignment and translation quality over already-powerful baselines on very large corpora.

2 A Feature-Rich Syntax-Aware Alignment Model

We follow Riesa and Marcu (2010) for efficient inference with arbitrary features, but do not rely upon hand-crafted syntactic patterns; rather, we extract syntactic features automatically from training data. We also introduce, in Section 5, an iterative approximate Viterbi inference procedure to deal with the asymmetry of the model. We show that this boosts both alignment and downstream translation quality even further.

The model itself is a linear combination of features, whose parameters are learned online via a structured perceptron (Collins, 2002). However, as we describe in Section 3, the features of the model are not known a priori. In what follows, we describe the search algorithm so that the reader has an understanding of the domain of locality before we begin to describe features and how they are learned.

2.1 Search Overview

We formulate the search for the best alignment as bottom-up parsing. Given a syntactic parse tree on one side of a parallel sentence, we use the structure of the tree to guide the search process. The key idea is that complex interactions between alignments are less likely to cross constituents, so we search recursively on the tree.

As an illustrative example, we point to the structure of the hypergraph search depicted in Figure 1. Here we are aligning the sentence pair:

a flag hung from the stage
台 上 挂 着 国旗
tái shàng guà zhe guóqí

The figure shows the search process for a small example with beam size *k*. Each black square represents a **partial alignment**. Each partial alignment at each node is ranked according to its model score. In this figure, the 1-best hypothesis at the leftmost NP node is constructed by composing the best hypothesis at its child DT and the 2nd-best hypothesis at its child NN. At the root node, we have a *k*-best list of full alignments.

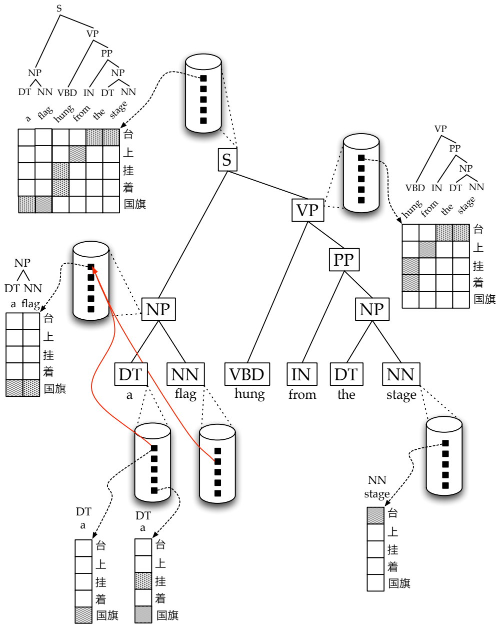


Figure 1: Approximate search through a hypergraph with beam size $k = 5$. Each black square represents a partial alignment; larger grey-shaded boxes are links in an alignment. Each partial alignment at each node is ranked according to its model score. The root node, S, contains a *k*-best list of full alignments.

We continue with a procedural description of the algorithm.

2.1.1 Initialization

We begin by visiting each preterminal node in sequence. We enumerate and score all one-to-one links as well as the unaligned link (aligned to null). Next, for a given preterminal node, we use cube pruning (Chiang, 2007) to find the top *k* one-to-two alignments, given the scores of the one-to-one links. We perform additional iterations of cube pruning to find top *k* sets of one-to-*m* links. In theory, we could increase *m* to the length of the foreign sentence and enumerate top *k* lists for each English word aligned to between 0 and all foreign words. However, in practice we set *m* to limit time spent here, while maintaining acceptable recall. In our experiments we set $m = 2$ for both English-Arabic and English-Chinese.

2.1.2 Combination

We continue traversing the tree bottom-up. At each nonterminal node, a *k*-best list of partial alignments from each of its child nodes are combined into a larger span. We use cube pruning to do this efficiently.[1] Nodes in different subtrees are processed independently of one another; i.e., for any node, alignment information at that node's sister is unavailable. For example, in Figure 1, alignment information at the leftmost NP is unavailable to us while we are constructing partial alignments at the PP. Search continues recursively up the tree, until we have reached the root node. The root node again computes the top *k* alignments from its children, and these comprise our final *k*-best list of full alignments.

In our experiments we only make use of the 1-best alignment for evaluation and translation. Previous work has shown that only shallow *k*-best lists of alignments may be beneficial, and that very deep *k*-best lists are not especially useful in improving final downstream translation grammar extraction due to rapid degradation in quality (Venugopal et al., 2008; Liu et al., 2009b); though they may have other uses.

3 Automatically Exploiting Syntactic Features for Alignment

Up to now, previous work in syntax-based alignment has largely modeled alignments based on features encoding target-side English syntactic and lexical information, but only lexical information on the source side.

However, there is much more data waiting to be exploited, and the flexible model and efficient and modular learning framework of hierarchical discriminative alignment afford us this possibility. Here, we discuss our target-side features, source-side features, and features that jointly take into account both source- and target-side information.

3.1 Target Syntax Features

Most alignment systems currently function without explicit regard to the downstream translation model. Some notable exceptions are May and Knight (2007) who generate syntactic alignments by re-aligning word-to-word alignments with a syntactic model; and Pauls and Klein (2010) who generate syntactic alignments with a synchronous ITG (Wu, 1997) approach. We depart from ITG-based models (Cherry and Lin, 2006; Haghighi et al., 2009) because of their complexity ($O(n^6)$ in the synchronous case), requiring heavy pruning or the computation of outside cost estimates (DeNero and Klein, 2010). Instead, we use linguistically motivated target-side parse trees to constrain search, as described above. These trees are output from the Berkeley parser (Petrov and Klein, 2007) and fixed at alignment time. We use these trees not only as a vehicle for search, but also for features.

A significant motivation for this work is the desire to make the connection, at alignment time, between translation rules used in decoding and the alignments that yield such translation rules. To do this, we fold the rule extraction process into the alignment search. At each step in the search process, we can extract translation rules from a given partial alignment and encode them as binary features.

Importantly, the rule extraction process itself is not directly tied to the alignment system, but rather to the downstream translation model. We can drop in any type of rule extraction we like into the alignment system, though some may generalize better than others to new data in a large corpus – important for supervised training conditions with relatively small amounts of annotated data.[2] In this work we focus on string-to-tree translation and the translation rule space described in (Galley et al., 2004; Galley et al., 2006).

During training and inference, we are constantly scoring partial alignments. Every time we have a partial alignment to score, we can extract all potential translation rules implied by that alignment, and encode those rules as features. In this case, we are doing two important things:

1. informing the alignment search with the rules of the translation model, and

2. modeling actual translation rules – the model parameters give us a way to quantify the relative importance of each rule.

For example, we learn that:

[1] Cube pruning is approximate when we have nonlocal combination features, and most of our features are of this type.

[2] For example, fully lexicalized phrase-based rules are less useful here than gapped phrases or hierarchical rules.

(1) Chinese VP and NP tend to be reordered around the 的 particle when translating to English.

feature	weight
NP(NP$_{\boxed{1}}$ VP$_{\boxed{2}}$) ↔ $\boxed{2}$ 的 $\boxed{1}$	1.01304

(2) When translating an Arabic NP as part of a VP, we often insert “is”.

feature	weight
VP((VBZ is) NP$_{\boxed{1}}$) ↔ $\boxed{1}$	0.67252

From this process we extract and learn 326, 239 lexicalized and non-lexicalized translation rule features in our Arabic-English model; 234, 972 in our Chinese-English model. Those features for which a positive weight is learned tend to generalize well over the training data; negatively weighted features do not, and are generally learned from alignments with mistakes during search. See Figure 2 for additional examples of rule features learned for Arabic-English alignment.

Negative evidence Nearly 67% of the rule features we learn for Chinese-English, and 55% of the rule features we learn for Arabic-English are negatively weighted. Early experiments involved only firing indicator rule features when an extracted rule at alignment-time matched in a set of rules extracted offline from our hand-aligned data. However, coverage from such rules will always be limited; firing every rule as a feature as it is encountered during search gives us many more darts to throw. Using only rule features extracted from gold data lowers F-measure by close to 5 points.

3.2 Source Syntax Features and Joint Features

Source syntactic trees have recently been shown to be helpful in machine translation decoding (Zhang et al., 2008; Liu et al., 2009a; Chiang, 2010), but to our knowledge have not been used in alignment models other than that of Burkett et al. (2010). We parse the source side of our data using the Berkeley parser (Petrov and Klein, 2007), and encode information provided by the source syntax as features in the model in two ways: (1) as tree-distance features[3], and (2) as joint source-target syntax features.

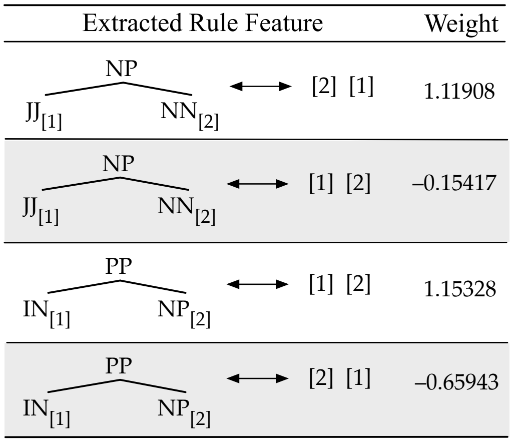


Figure 2: Translation rules as features extracted during Arabic-English alignment. These rules show that we learn to reorder adjectives and nouns inside noun phrases, and that prepositions before sister NPs prefer to be translated monotonically. For Chinese-English, we learn the opposite.

3.2.1 Source-Target Coordination Features

Drawing on work by Chiang (2010) in stochastically rewriting syntactic constituents across languages in a translation model, we adapt the general idea to alignment modeling. Chiang calls these features *fuzzy syntax* features; here, we simply call them *coordination* features in our adaptation for alignment, so as to avoid the implication that we are rewriting.

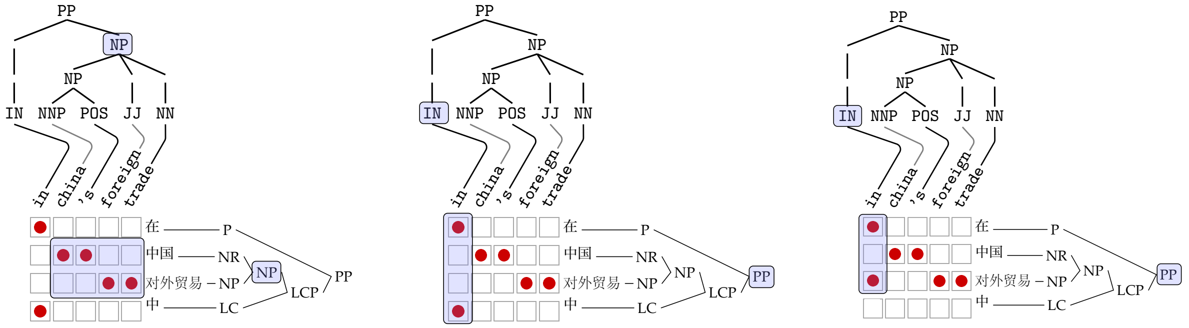
This feature family is a set of binary features that may fire at any nonterminal node in the tree during bottom-up search. A feature fires for each combination of two nonterminal source and target nodes s and t, respectively, that match the following conditions:

1. t is the label of the current target tree node in the bottom-up search.

2. s is the label of the source tree node of maximal depth (i.e. closest to leaf nodes) that spans all links also spanned by t.

Figure 3 shows three examples of this joint feature over source and target trees. In Figure 3a, the maximal-depth source tree node that spans every link also spanned by the shaded target tree NP

[3] These features parameterize the intuition that if two source words align to a single target word, we prefer them to be members of the same constituent, or having a short path through the tree from one word to the other, e.g. (in, 在...中), or the first and last Chinese words in the examples in Figure 3.



(a) Source/target tree feature firing at node NP, with value ⟨ NP ; NP ⟩. The maximal-depth source tree node that spans every link also spanned by the shaded target tree NP is also labeled NP.

(b) Source/target tree feature firing at node IN, returning value ⟨ IN ; PP ⟩.

(c) In this figure, depicting an incorrect alignment, the same feature value is fired as for the correct alignment in 3b: ⟨ IN ; PP ⟩. We need more contextual annotation to create more discriminative power.

Figure 3: Two examples of joint features over monolingual parse trees. The value of the feature depends on the shaded areas.

is also labeled NP. So, the feature returns a value of ⟨NP; NP⟩. In Figure 3b, PP is the label of the maximal-depth source tree node that spans every link also spanned by the shaded target tree IN node; the feature fires a value ⟨IN ; PP⟩. We might expect this pairing of IN with PP, or of IN with P, but we would expect to learn a penalizing parameter weight for the pairing of, say, IN with NP.

Adding more context Powerful as this feature is, it is not quite discriminating enough; it may return the same feature value for both a correct and incorrect alignment, as shown in Figure 3c. To overcome this, we introduce additional features annotated with the left-most and right-most tags in the current span. For example, in this figure, we also fire ⟨ IN ; PP(P,NP) ⟩, and learn a negative weight of −0.638 denoting a poor choice of alignment. We also find it helpful to keep the original unannotated feature as a poor-man's backoff.

Some examples Table 1 shows some of the maxmially and minimally-weighted features learned. As the more highly weighted features show, both models learn to prefer alignments that result in the coordination of similar constituent labels. For example, the Chinese model learns a very high weight for aligning sets of English words that form prepositional phrases to sets of Chinese words that also form prepositional phrases[4].

	Ara-Eng Model			Chi-Eng Model		
	eng	ara	w	eng	chi	w
[1]	SBAR	SBAR	6.40	PP	PP	10.3
[2]	S	S(CC,PU)	4.91	NP	NP	9.38
[3]	PP	PP	4.20	SBAR	VP(VV,PU)	6.97
[4]	VP	VP	3.90	NP	NP(DT,NN)	6.67
[5]	SBAR	PP	2.58	PP	PP(P,LC)	6.38
[6]	NP	S	-2.80	NP	PP	-6.82
[7]	NP	VP	-3.01	S	IP(PU,PU)	-7.44
[8]	NP	NP(NN,IN)	-4.52	PP	IP	-7.33
[9]	PP	VP	-5.13	SBAR	VP	-7.72
[10]	PP	S	-7.37	NP	IP	-7.83

Table 1: This table shows a sampling of the highest and lowest-weighted coordination features applied when scoring partial alignments at nodes in the tree. Preterminal tags inside parentheses indicate the POS tags on the left and right edge of a given constituent.

Inversely, we learn high negative weights for model features that fire for alignments that oblige the firing of features of very dissimilar nonterminal labels, and that often yield asynchronous bracketing. For example, the Arabic model learns that English words that form prepositional phrases should

[4] In Table 1, Chinese feature [1].

not align to sets of Arabic words that form entire sentences or verb phrases[5].

In total, we learn 127,932 syntactic coordination features in our Arabic-English model; 59,239 for Chinese-English.

4 Learning

We learn feature weights using a parallelized implementation of online averaged perceptron (Collins, 2002). We distribute training examples to CPUs in a cluster and essentially run several perceptron learners in parallel. We communicate and average the weight vectors of each learner according to the Iterative Parameter Mixing strategy described by McDonald et al. (2010).

At each iteration, our perceptron update is:

$$\mathbf{w} \leftarrow \mathbf{w} + \mathbf{h}(y_i) - \mathbf{h}(\hat{y}) \tag{3}$$

And we define:

$$\hat{y} = \underset{y \in \text{cand}(x)}{\arg\max}\; \ell(y_i, y) + \mathbf{w} \cdot \mathbf{h}(y) \tag{4}$$

$$\ell(y_i, y) = 1 - F_1(y_i, y) \tag{5}$$

with $\mathbf{w}$ our weight vector, $\mathbf{h}(y)$ our sparse vector of feature values, and $F_1(y_i, y)$ balanced F-measure. The loss, $\ell(y_i, y)$, is a measure of how bad it would be to guess $\hat{y}$ instead of y.

In selecting $\hat{y}$, we draw upon the loss-augmented inference literature (Tsochantaridis et al., 2004; Taskar et al., 2005a). Alignment $\hat{y}$ is the output candidate maximizing the sum of both the loss and model score. This guess appears attractive to the model, yet has low F-measure, and so is exactly the sort of output we would like to update away from.

During training, we learn both the parameters and model structure. Figure 4b shows how the size of the model grows over time. As described in Sections 2 and 3, we automatically extract and fire features given an alignment configuration and our current position in the tree. We see a steep initial growth in model size, and then begin to trail off as the number of new unique rules and negative evidence we encounter diminishes.

Model Selection Among models from the first iteration up to convergence, we choose the model parameters from the best performing model as measured by F-measure on a held-out development set of alignments.

5 Iterative Approximate Viterbi Inference

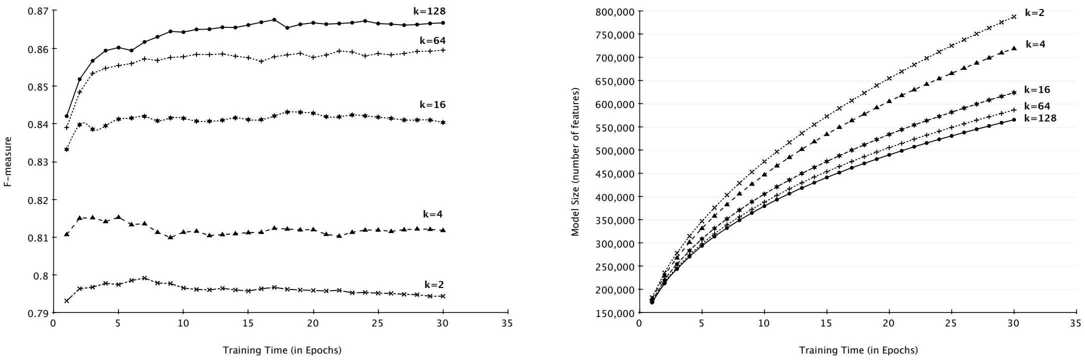
Though up to now we have described features that fire during bottom-up search on the target-language tree, we can also search bottom-up on the source-language tree. The syntactic features we have described are generic enough that they will still be extractable and applicable. Because our model and inference procedure are asymmetric, a search on the source-language tree will generate alignments from a different space, and can provide a unique signal we would not otherwise have. We can use the Viterbi alignments from each model to inform the other. In the following we describe a method for simultaneously training both target-tree and source-tree models but with features to enforce agreement, somewhat similar to (Nivre and McDonald, 2008) in integrating two dependency parsing models.

We begin by training two models, one that operates on the target tree, and one that operates on the source tree. Call the parameters learned from these models $\mathbf{w}_1^t$ and $\mathbf{w}_1^s$, respectively. Then, performing inference under these models yields alignments a_1^t and a_1^s.

In the next iteration we learn parameters $\mathbf{w}_2^t$ and $\mathbf{w}_2^s$, and introduce agreement features. In this step, during training to find $\mathbf{w}_2^t$, the target-tree model uses a_1^s to fire indicator features. These fire for any alignment link that was also present in the previous iteration's source-tree alignment, a_1^s. Analogously, when searching for the best $\mathbf{w}_2^s$, we use a_1^t to fire indicator features that fire for any alignment link also present in the previous iteration's target-tree alignment, a_1^t.

This process of using the alignment from the previous iteration's opposing tree continues until convergence, i.e. until we no longer see improvement in our 1-best source-tree and target-tree alignments. When we use these alignments for downstream translation, we symmetrize with the *grow-diag-final* heuristic, which continues to work remarkably well in practice. We also experiment with the intersection of both final alignments.

[5] In Table 1, Arabic features [6] and [7].



(a) Learning curves (Arabic-English): F-measure accuracy on heldout development data over time for five different beam settings, *k*=2, *k*=4, *k*=16, *k*=64 and *k*=128. For Arabic-English, improvements are minimal with beams larger than *k*=128; and for Chinese-English, with beams larger than *k*=256.

(b) Model size as a function of time for five different beam settings (Arabic-English): We see a steep initial growth, and then begin to trail off as the number of new unique extractable features and negative evidence we encounter diminishes. Growth rate is higher for models with narrower beams that make more mistakes.

Figure 4: Learning feature-rich alignment models. Figure 4a shows learning curves on heldout data for five different beam sizes. Figure 4b shows how the models dynamically grow over time. In Figure 4b we notice that less accurate models with narrower beams need to add more complexity in an attempt to make up for their many more mistakes.

6 Evaluation

6.1 Alignment Quality

From LDC2006E86 and LDC2006E83, we use as training data 2,280 hand-aligned sentence pairs of Arabic-English and 1,102 for Chinese-English. We measure training convergence using a held-out development set of 100 sentence pairs for each language pair, and evaluate with F-measure on a held-out test set of 184 sentences pairs for Chinese-English and 364 sentence pairs for Arabic-English. We use instances of the Berkeley parser (Petrov and Klein, 2007) trained on the English Penn Treebank, Chinese Treebank 6, and the Arabic Treebank parts 1–3; for each language, trees are fixed at alignment time using the 1-best output from each parser.

We use Model-4 symmetrized with the grow-diag-final heuristic, trained with GIZA++ as a baseline alignment model. We train two GIZA++ models on our largest available Chinese-English and Arabic-English parallel corpora. These consist of 261M and 223M English words,[6] respectively. The size of these corpora make for quite a powerful unsupervised baseline.

In training our alignment model, we use the syntactic features discussed in Section 3, plus word-based lexical features $t(e \mid f)$ and $t(f \mid e)$ used during initialization, extracted offline directly from the translation-table of GIZA++. Using these features alone results in an F-measure of 59.1 for Arabic-English, and 55.6 for Chinese-English. Our automatically extracted syntactic features and iterative inference algorithm get us the rest of the way, bringing performance up to 87.6 and 87.0, respectively.

Table 2 shows the results on our held-out 100-sentence test set. In an intrinsic evaluation on an alignment task, our F-measure scores are more than 15 points higher than the baseline for both language pairs.

6.2 Translation Quality

In evaluating downstream translation quality, we build three translation systems each for Arabic-English and Chinese-English: one with alignments from GIZA++, one with alignments from our syntactically-informed discriminative model, and one with alignments from our model with iterative inference (Section 5). For each of these systems we

[6] These counts correspond to 240M words of Chinese and 194M words of Arabic.

	Arabic-English			Chinese-English		
	F	P	R	F	P	R
GIZA++ M4 grow-diag-final	72.5	74.5	70.5	71.7	71.4	72.0
Target-tree alignments only	86.8	89.1	84.6	84.4	89.4	80.0
with Iterative Inference (grow-diag-final)	**87.6**	89.7	**85.6**	**87.0**	90.0	**84.1**
with Iterative Inference (intersection)	83.4	**93.1**	75.6	83.1	**95.4**	73.6

Table 2: F-measure, Precision, Recall for GIZA++ Model-4, and for alignments from this work. GIZA++ was trained on 223M words for Arabic-English, and 261M words for Chinese-English. We observe very large gains in accuracy of 15 points for both language pairs. Iterative inference with source and target-tree alignments yields a large effect on Chinese-English recall, and a modest improvement in Arabic-English.

align our parallel training corpora described in Section 6.1, and compute word-based lexical weighting features (Koehn et al., 2003) based on these alignments.

Because of the number of experiments involved in this research, we needed to accelerate our downstream experimental pipeline. While we align our full training corpus, we extract translation rules from a subset of our alignment training data; the quality of the translation rules extracted is still a function of the original alignment model.

We train a syntax-based string-to-tree translation model (Galley et al., 2004; Galley et al., 2006) and extract translation rules.[7] using alignments produced by each system from 4.25+5.43M words for Arabic-English and 31.8+37.7M words for Chinese-English. For Arabic-English, we tune our MT system on a held-out development corpus of 1,172 parallel sentences, and test on a heldout set of 746 parallel sentences with four references each. For Chinese-English we tune our MT system on a held-out development corpus of 4,089 parallel sentences, and test on a set of 4,060 sentences with four references each. We tune the translation models for these systems with MIRA (Watanabe et al., 2007; Chiang et al., 2008). Our tuning and test corpora are drawn from the NIST 2004 and 2006 evaluation data, disjoint from our rule-extraction data. All systems used two language models; one trained on the combined English sides of our Arabic-English and Chinese-English data (480M words), and one trained on 4 billion words of English data.

MT results are shown in Table 3. We show a gain of 1.0 and 1.3 BLEU points over GIZA++ Model-4. Each is statistically significant over the baseline.

Alignment model	ara-eng **BLEU**	chi-eng **BLEU**
GIZA++ Model-4	47.6	26.2
Target-tree alignments only	48.3*	26.4$^+$
+Iterative Inference (gdf)	48.4	27.0*
+Iterative Inference (intersection)	**48.6**$^+$	**27.5***

Table 3: IBM BLEU scores using a syntax-based MT system. We show statistically significant gains in both language pairs over unsupervized GIZA++ Model 4 trained on very large corpora. An asterisk (*) denotes a statistically significant improvement with $p < 0.01$ over the number immediately above; a (+) denotes $p < 0.05$.

In the case of Chinese-English, we see a 1.1 BLEU gain when using iterative inference over the standard model which provides only target-tree alignments. As measured by a bootstrap resampler, this improvement is statistically significant, with $p < 0.01$.

For Arabic-English, we see a BLEU gain of 0.7 with target-tree alignments alone, and a total 1.0 BLEU gain over the baseline with iterative inference and our joint-agreement features.

We expect the limited improvement of iterative inference for Arabic-English is due to at least two factors:

1. the relative weakness of our Arabic parser, and
2. as shown in Table 2, our Arabic target-tree alignments are already quite accurate.

[7] We use the so-called *composed* rules of (Galley et al., 2006).

7 Discussion

We achieve our best downstream BLEU results when using iterative inference with source-tree and target-tree alignments, keeping the intersection.[8] These alignments have been shown to have recall in a similar neighborhood as our unsupervised baseline, but extremely high precision.

As DeNero and Klein (2010) and others have observed, the relationship between word alignment evaluation metrics and BLEU score remains tenuous at best. While we are able to induce some of the most accurate alignments we have seen to date, it remains unclear, given our gold hand-aligned data, whether we are optimizing for the right function ultimately for the translation task. Related metrics, like Rule F-measure (Fossum et al., 2008) and Translation Unit Error Rate (Søgaard and Kuhn, 2009), are still functions of a given gold alignment. If the gold alignment is not ideally annotated for the translation task, it matters little what our alignment evaluation metric is.

Why do grow-diag-final alignments (for our system) not perform as well? We believe the answer lies in the fact that these alignments *too closely* resemble the gold alignments with word-alignment annotation standards[9] that do not handle function words ideally for the translation task. Indeed, Hermjakob (2009) reports improved BLEU with a hand-modified gold standard.

Interestingly, the places in which our source-tree and target-tree alignments most often disagree is in the alignment of function words with no clear translation in the opposite language. For example, English *the* has no translation in Chinese. Our intersection alignments generally leave *the* unaligned to Chinese words, whereas in our gold alignments *the* is generally aligned to the same word as the head of the NP in which it appears.[10]

We see our best translation performance with our intersection alignments because we believe it largely leaves untranslated words and words without clear translations in the opposite language unaligned; we believe this may be the right thing to do.[11] Continuing with the *the* example, our translation model learns to insert words like *the* where appropriate, and such insertion rules are validated by the language model. We learn with good coverage accurate high-precision translation rules for content words, and general insertion rules for words like *the*, instead of learning two unique lexicalized rules for a given content word, one with and one without *the*. In this way, we are learning a more general grammar that explains the data.

8 Conclusion

In this work we are closing the gap between translation and alignment models in terms of syntactic sophistication. We have (1) shown how to efficiently extract hundreds of thousands of language-independent syntactic features useful for alignment, (2) given a detailed analysis of the types of linguistic phenomena these varied features generalize, and (3) report significant gains not only on alignment quality but also on downstream machine translation quality (1.0+ BLEU) over very strong baselines across diverse language pairs.

We have also hinted at roadblocks to improved discriminative alignment modeling for translation. We expect that an accurate discriminative word alignment system, such as the one presented here, in conjunction with better annotation standards for alignment will take us even farther beyond the advancements in translation quality shown here.

Acknowledgements

The authors would like to thank David Chiang, Steve DeNeefe, Liang Huang, Kevin Knight, Jonathan May, and the anonymous reviewers for their thoughtful comments. This work was supported in part by NSF IIS-0908532, DARPA contract HR0011-06-C-0022 under subcontract to BBN Technologies, and a USC CREATE Fellowship to the first author.

[8] Intersection symmetrization does not help GIZA++ because the resulting recall is so low as to severely limit the usefulness of direct translation rule extraction with such alignments (49.7 Recall for Chi-Eng; 47.2 Recall for Ara-Eng).

[9] We refer to those used for data used in this work, LDC2006E86 and LDC2006E93, as well as the standards for later hand-aligned data developed for the GALE program.

[10] E.g., ⟨(the country , 国家)⟩; but not, ⟨(the, ∅); (country, 国家)⟩

[11] Naively leaving all function words unaligned is likely sub-optimal, as many have seem to have direct translations in some contexts; cf. (of, من) and (of, 的).

References

Phil Blunsom and Trevor Cohn. 2006. Discriminative word alignment with conditional random fields. In *Proceedings of COLING-ACL*, pages 65–72, Sydney, Australia.

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

David Burkett, John Blitzer, and Dan Klein. 2010. Joint parsing and alignment with weakly synchronized grammars. In *Proceedings of NAACL HLT 2010*, pages 127–135, Los Angeles, CA. USA.

Colin Cherry and Dekang Lin. 2006. Soft syntactic constraints for word alignment through discriminative training. In *Proceedings of COLING/ACL*, pages 105–112, Sydney, Australia. Association for Computational Linguistics.

David Chiang, Yuval Marton, and Philip Resnik. 2008. Online large-margin training of syntactic and structural translation features. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 224–233, Honolulu, HI. USA.

David Chiang. 2007. Hierarchical phrase-based translation. *Computational Linguistics*, 33(2):201–228.

David Chiang. 2010. Learning to translate with source and target syntax. In *Proceedings of the 48th Annual Meeting of the ACL*, pages 1443–1452, Uppsala, Sweden.

Michael Collins. 2002. Discriminative training methods for hidden markov models: theory and experiments with perceptron algorithms. In *Proceedings of EMNLP*, pages 1–8, Philadelphia, PA. USA.

John DeNero and Dan Klein. 2007. Tailoring word alignments to syntactic machine translation. In *Proceedings of the 45th annual meeting of the ACL*, pages 17–24, Prague, Czech Republic.

John DeNero and Dan Klein. 2010. Discriminative modeling of extraction sets for machine translation. In *Proceedings of NAACL HLT 2010*, pages 1453–1463, Los Angeles, CA. USA.

Victoria Fossum, Kevin Knight, and Steven Abney. 2008. Using syntax to improve word alignment for syntax-based statistical machine translation. In *Proceedings of ACL MT Workshop*, pages 44–52, Honolulu, HI. USA.

Alexander Fraser and Daniel Marcu. 2007. Getting the structure right for word alignment: LEAF. In *Proceedings of EMNLP-CoNLL*, pages 51–60, Prague, Czech Republic.

Michel Galley, Mark Hopkins, Kevin Knight, and Daniel Marcu. 2004. What's in a translation rule? In *Proceedings of HLT-NAACL*, pages 273–280.

Michel Galley, Jonathan Graehl, Kevin Knight, Daniel Marcu, Steve DeNeefe, Wei Wang, and Ignacio Thayer. 2006. Scalable inference and training of context-rich syntactic translation models. In *Proceedings of COLING-ACL*, pages 961–968, Sydney, Australia. Association for Computational Linguistics.

Aria Haghighi, John Blitzer, John DeNero, and Dan Klein. 2009. Better word alignments with supervised itg models. In *Proceedings of the Joint Conference of the ACL and IJCNLP*, pages 923–931, Singapore, August.

Ulf Hermjakob. 2009. Improved word alignment with statistics and linguistic heuristics. In *Proceedings of EMNLP*, pages 229–237, Singapore.

Liang Huang and Haitao Mi. 2010. Efficient incremental decoding for tree-to-string translation. In *Proceedings of EMNLP 2010*, pages 273–283, Boston, MA. USA.

Abraham Ittycheriah and Salim Roukos. 2005. A maximum entropy word aligner for Arabic-English machine translation. In *Proceedings of HLT-EMNLP*, pages 89–96, Vancouver, Canada.

Philipp Koehn, Franz J. Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of HLT-NAACL*, pages 127–133, Edmonton, Canada.

Simon Lacoste-Julien, Dan Klein, Ben Taskar, and Michael Jordan. 2006. Word alignment via quadratic assignment. In *Proceedings of HLT-NAACL*, pages 112–119, New York, NY. USA.

Zhifei Li, Chris Callison-Burch, Chris Dyer, Juri Ganitkevitch, Sanjeev Khudanpur, Lane Schwartz, Wren Thornton, Jonathan Weese, and Omar Zaidan. 2009. Joshua: An open source toolkit for parsing-based machine translation. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 135–139, Athens, Greece.

Yang Liu, Qun Liu, and Shouxun Lin. 2005. Log-linear models for word alignment. In *Proceedings of the 43rd annual meeting of the ACL*, pages 459–466, Ann Arbor, MI.

Yang Liu, Yajuan Lü, and Qun Liu. 2009a. Improving tree-to-tree translation with packed forests. In *Proceedings of the 47th Annual Meeting of the ACL and the 4th IJCNLP of the AFNLP*, pages 558–566, Singapore.

Yang Liu, Tian Xia, Xinyan Xiao, and Qun Liu. 2009b. Weighted alignment matrices for statistical machine translation. In *Proceedings of EMNLP*, pages 1017–1026, Singapore.

Jonathan May and Kevin Knight. 2007. Syntactic re-alignment models for machine translation. In *Proceedings of EMNLP*, pages 360–368, Prague, Czech Republic.

Ryan McDonald, Keith Hall, and Gideon Mann. 2010. Distributed training strategies for the structured perceptron. In *Proceedings of NAACL HLT*, pages 456–464, Los Angeles, CA. USA.

Robert C. Moore, Wen-Tau Yih, and Andreas Bode. 2006. Improved discriminative bilingual word alignment. In *Proceedings of COLING-ACL*, pages 513–520, Sydney, Australia.

Robert C. Moore. 2005. A discriminative framework for bilingual word alignment. In *Proceedings of HLT-EMNLP*, pages 81–88, Vancouver, Canada.

Joakim Nivre and Ryan McDonald. 2008. Integrating graph-based and transition-based dependency parsers. In *Proceedings of the 46th Annual Meeting of the ACL*, pages 950–958, Columbus, OH. USA.

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

Adam Pauls and Dan Klein. 2010. Unsupervised syntactic alignment with inversion transduction grammars. In *Proceedings of NAACL HLT 2010*, pages 118–126, Los Angeles, CA. USA.

Slav Petrov and Dan Klein. 2007. Improved inference for unlexicalized parsing. In *Proceedings of NAACL HLT*, pages 404–411, Rochester, NY. USA.

Jason Riesa and Daniel Marcu. 2010. Hierarchical search for word alignment. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 157–166, Uppsala, Sweden.

Anders Søgaard and Jonas Kuhn. 2009. Empirical lower bounds on alignment error rates in syntax-based machine translation. In *SSST '09: Proceedings of the Third Workshop on Syntax and Structure in Statistical Translation*, pages 19–27. Association for Computational Linguistics.

Ben Taskar, Vassil Chatalbashev, Daphne Koller, and Carlos Guestrin. 2005a. Learning structured prediction models: A large margin approach. In *Proceedings of ICML*, pages 896–903, Bonn, Germany.

Ben Taskar, Simon Lacoste-Julien, and Dan Klein. 2005b. A discriminative matching approach to word alignment. In *Proceedings of HLT-EMNLP*, pages 73–80, Vancouver, Canada.

Tsochantaridis, Thomas Hofmann, Thorsten Joachims, and Yasemin Altun. 2004. Support vector machine learning for interdependent and structured output spaces. In *Proceedings of ICML*, Banff, AB. Canada.

Ashish Venugopal, Andreas Zollmann, Noah A. Smith, and Stephan Vogel. 2008. Wider pipelines: *N*-best alignments and parses in MT training. In *Proceedings of AMTA*, pages 192–201, Honolulu, HI. USA.

Taro Watanabe, Jun Suzuki, Hajime Tsukuda, and Hideki Isozaki. 2007. Online large-margin training for statistical machine translation. In *Proceedings of EMNLP*, pages 764–773.

Dekai Wu. 1997. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Computational Linguistics*, 23(3):377–403.

Min Zhang, Hongfei Jiang, Aiti Aw, Haizhou Li, Chew Lim Tan, and Sheng Li. 2008. A tree sequence alignment-based tree-to-tree translation model. In *Proceedings of ACL-08: HLT*, pages 559–567, Columbus, OH. USA.

Andreas Zollman and Ashish Venugopal. 2006. Syntax augmented machine translation via chart parsing. In *NAACL 2006 Workshop on Statistical Machine Translation*, pages 138–141, Rochester, NY. USA.