Using Syntax to Improve Word Alignment Precision for Syntax-Based Machine Translation

Victoria Fossum
Dept. of Computer Science
University of Michigan
Ann Arbor, MI 48104
vfossum@umich.edu

Kevin Knight
Information Sciences Institute
University of Southern California
Marina del Rey, CA 90292
knight@isi.edu

Steven Abney
Dept. of Linguistics
University of Michigan
Ann Arbor, MI 48104
abney@umich.edu

Abstract

Word alignments that violate syntactic correspondences interfere with the extraction of string-to-tree transducer rules for syntax-based machine translation. We present an algorithm for identifying and deleting incorrect word alignment links, using features of the extracted rules. We obtain gains in both alignment quality and translation quality in Chinese-English and Arabic-English translation experiments relative to a GIZA++ union baseline.

1 Introduction

1.1 Motivation

Word alignment typically constitutes the first stage of the statistical machine translation pipeline. GIZA++ (Och and Ney, 2003), an implementation of the IBM (Brown et al., 1993) and HMM (**?**) alignment models, is the most widely-used alignment system. GIZA++ *union* alignments have been used in the state-of-the-art syntax-based statistical MT system described in (Galley et al., 2006) and in the hierarchical phrase-based system Hiero (Chiang, 2007). GIZA++ *refined* alignments have been used in state-of-the-art phrase-based statistical MT systems such as (Och, 2004); variations on the refined heuristic have been used by (Koehn et al., 2003) (*diag* and *diag-and*) and by the phrase-based system Moses (*grow-diag-final*) (Koehn et al., 2007).

GIZA++ union alignments have high recall but low precision, while *intersection* or refined alignments have high precision but low recall.[1] There are two natural approaches to improving upon GIZA++ alignments, then: deleting links from union alignments, or adding links to intersection or refined alignments. In this work, we delete links from GIZA++ union alignments to improve precision.

The low precision of GIZA++ union alignments poses a particular problem for syntax-based rule extraction algorithms such as (Quirk et al., 2005; Galley et al., 2006; Huang et al., 2006; Liu et al., 2006): if the incorrect links violate syntactic correspondences, they force the rule extraction algorithm to extract rules that are large in size, few in number, and poor in generalization ability.

Figure 1 illustrates this problem: the dotted line represents an incorrect link in the GIZA++ union alignment. Using the rule extraction algorithm described in (Galley et al., 2004), we extract the rules shown in the leftmost column (R1–R4). Rule R1 is large and unlikely to generalize well. If we delete the incorrect link in Figure 1, we can extract the rules shown in the rightmost column (R2–R9): Rule R1, the largest rule from the initial set, disappears, and several smaller, more modular rules (R5–R9) replace it.

In this work, we present a supervised algorithm that uses these two features of the extracted rules (size of largest rule and total number of rules), as well as a handful of structural and lexical features, to automatically identify and delete incorrect links from GIZA++ union alignments. We show that link

[1] For a complete discussion of alignment symmetrization heuristics, including union, intersection, and refined, refer to (Och and Ney, 2003).

Proceedings of the Third Workshop on Statistical Machine Translation, pages 44–52,
Columbus, June 2008.

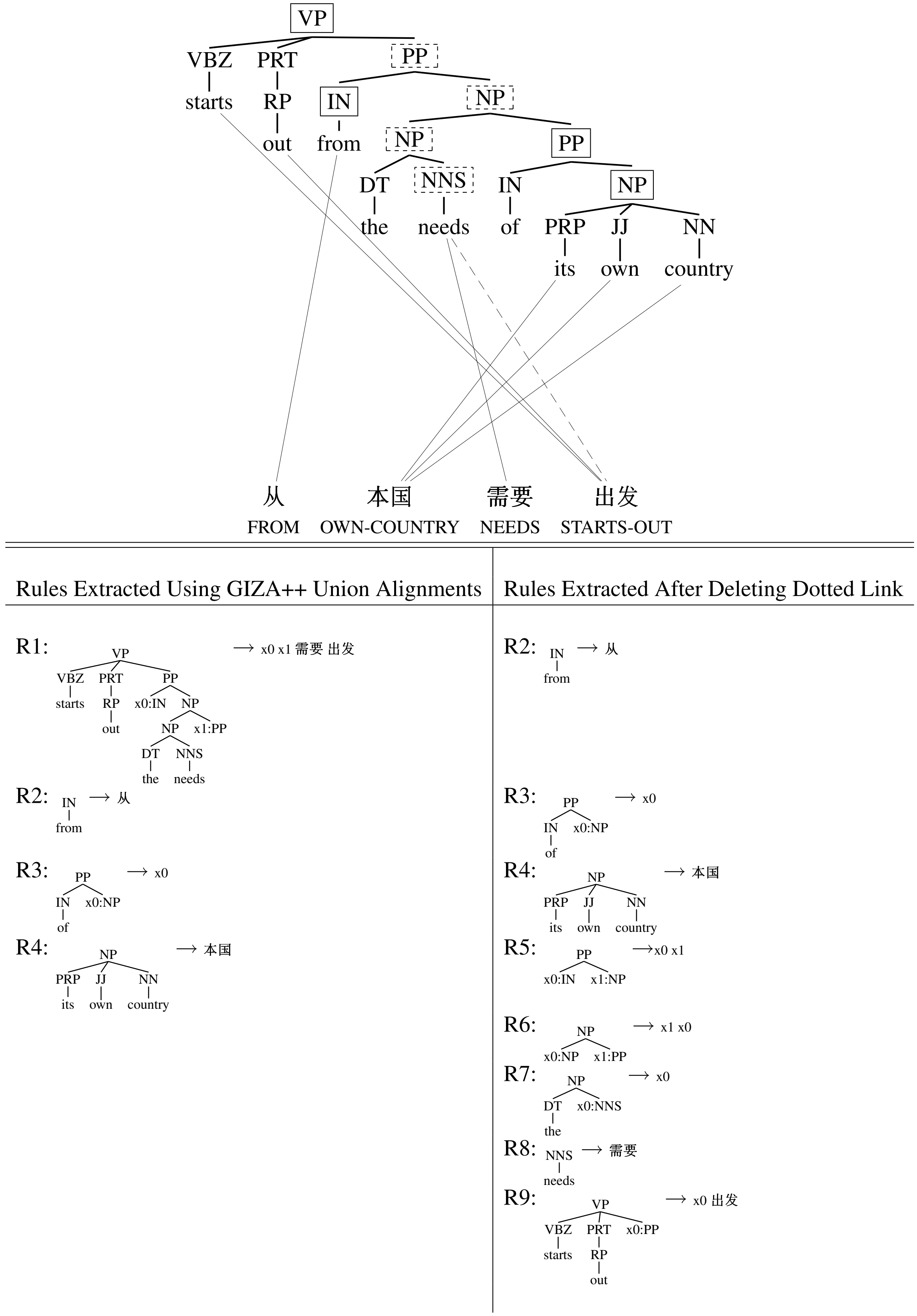

Figure 1: The impact of incorrect alignment links upon rule extraction. Using the original alignment (including all links shown) leads to the extraction of the tree-to-string transducer rules whose left hand sides are rooted at the solid boxed nodes in the parse tree (R1, R2, R3, and R4). Deleting the dotted alignment link leads to the omission of rule R1, the extraction of R9 in its place, the extraction of R2, R3, and R4 as before, and the extraction of additional rules whose left hand sides are rooted at the dotted boxed nodes in the parse tree (R5, R6, R7, R8).

deletion improves alignment quality and translation quality in Chinese-English and Arabic-English MT, relative to a strong baseline. Our link deletion algorithm is easy to implement, runs quickly, and has been used by a top-scoring MT system in the Chinese newswire track of the 2008 NIST evaluation.

1.2 Related Work

Recently, discriminative methods for alignment have rivaled the quality of IBM Model 4 alignments (Liu et al., 2005; Ittycheriah and Roukos, 2005; Taskar et al., 2005; Moore et al., 2006; Fraser and Marcu, 2007b). However, except for (Fraser and Marcu, 2007b), none of these advances in alignment quality has improved translation quality of a state-of-the-art system. We use a discriminatively trained model to identify and delete incorrect links, and demonstrate that these gains in alignment quality lead to gains in translation quality in a state-of-the-art syntax-based MT system. In contrast to the semi-supervised LEAF alignment algorithm of (Fraser and Marcu, 2007b), which requires 1,500-2,000 CPU *days* per iteration to align 8.4M Chinese-English sentences (anonymous, p.c.), link deletion requires only 450 CPU *hours* to re-align such a corpus (after initial alignment by GIZA++, which requires 20-24 CPU days).

Several recent works incorporate syntactic features into alignment. (May and Knight, 2007) use syntactic constraints to re-align a parallel corpus that has been aligned by GIZA++ as follows: they extract string-to-tree transducer rules from the corpus, the target parse trees, and the alignment; discard the initial alignment; use the extracted rules to construct a forest of possible string-to-tree derivations for each string/tree pair in the corpus; use EM to select the Viterbi derivation tree for each pair; and finally, induce a new alignment from the Viterbi derivations, using the re-aligned corpus to train a syntax-based MT system. (May and Knight, 2007) differs from our approach in two ways: first, the set of possible re-alignments they consider for each sentence pair is limited by the initial GIZA++ alignments seen over the training corpus, while we consider all alignments that can be reached by deleting links from the initial GIZA++ alignment for that sentence pair. Second, (May and Knight, 2007) use a time-intensive training algorithm to select the best re-alignment for each sentence pair, while we use a fast greedy search to determine which links to delete; in contrast to (May and Knight, 2007), who require 400 CPU hours to re-align 330k Chinese-English sentence pairs (anonymous, p.c), link deletion requires only 18 CPU hours to re-align such a corpus.

(Lopez and Resnik, 2005) and (Denero and Klein, 2007) modify the distortion model of the HMM alignment model (Vogel et al., 1996) to reflect tree distance rather than string distance; (Cherry and Lin, 2006) modify an ITG aligner by introducing a penalty for induced parses that violate syntactic bracketing constraints. Similarly to these approaches, we use syntactic bracketing to constrain alignment, but our work extends beyond improving alignment quality to improve translation quality as well.

2 Link Deletion

We propose an algorithm to re-align a parallel bitext that has been aligned by GIZA++ (IBM Model 4), then symmetrized using the union heuristic. We then train a syntax-based translation system on the re-aligned bitext, and evaluate whether the re-aligned bitext yields a better translation model than a baseline system trained on the GIZA++ union aligned bitext.

2.1 Link Deletion Algorithm

Our algorithm for re-alignment proceeds as follows. We make a single pass over the corpus. For each sentence pair, we initialize the alignment $A = A_{initial}$ (the GIZA++ union alignment for that sentence pair). We represent the score of A as a weighted linear combination of features h_i of the alignment A, the target parse tree $parse(e)$ (a phrase-structure syntactic representation of e), and the source string f:

$$score(A) = \sum_{i=0}^{n} \lambda_i \cdot h_i(A, parse(e), f)$$

We define a *branch* of links to be a *contiguous* 1-to-many alignment.[2] We define two alignments, A

[2] In Figure 1, the 1-to-many alignment formed by {本国-its, 本国- own,本国-country} constitutes a branch, but the 1-to-many alignment formed by {出发-starts,出发-out,出发-needs} does not.

and A', to be *neighbors* if they differ only by the deletion of a link or *branch* of links. We consider all alignments A' in the *neighborhood* of A, greedily deleting the link l or branch of links b maximizing the score of the resulting alignment $A' = A \setminus l$ or $A' = A \setminus b$. We delete links until no further increase in the score of A is possible.[3]

In section 2.2 we describe the features h_i, and in section 2.4 we describe how to set the weights λ_i.

2.2 Features

2.2.1 Syntactic Features

We use two features of the string-to-tree transducer rules extracted from A, $parse(e)$, and f according to the rule extraction algorithm described in (Galley et al., 2004):

ruleCount: Total number of rules extracted from A, $parse(e)$, and f. As Figure 1 illustrates, incorrect links violating syntactic brackets tend to decrease **ruleCount**; **ruleCount** increases from 4 to 8 after deleting the incorrect link.

sizeOfLargestRule: The size, measured in terms of internal nodes in the target parse tree, of the single largest rule extracted from A, $parse(e)$, and f. In Figure 1, the largest rules in the leftmost and rightmost columns are R1 (with 9 internal nodes) and R9 (with 4 internal nodes), respectively.

2.2.2 Structural Features

wordsUnaligned: Total number of unaligned words.

1-to-many Links: Total number of links for which one word is aligned to multiple words, in either direction. In Figure 1, the links {出发-starts,出发-out,出发-needs} represent a 1-to-many alignment. 1-to-many links appear more frequently in GIZA++ union alignments than in gold alignments, and are therefore good candidates for deletion. The category of 1-to-many links is further subdivided, depending on the degree of *contiguity* that the link exhibits with its neighbors.[4] Each link in a 1-to-many alignment can have 0, 1, or 2 neighbors, according to how many links are adjacent to it in the 1-to-many alignment:

zeroNeighbors: In Figure 1, the link 出发-needs has 0 neighbors.

oneNeighbor: In Figure 1, the links 出发-starts and 出发-out each have 1 neighbor–namely, each other.

twoNeighbors: In Figure 1, in the 1-to-many alignment formed by {本国-its,本国-own,本国-country}, the link 本国-own has 2 neighbors, namely 本国-it and 本国-country.

2.2.3 Lexical Features

highestLexProbRank: A link e_i-f_j is "max-probable from e_i to f_j" if $p(f_j|e_i) > p(f_{j'}|e_i)$ for all alternative words $f_{j'}$ with which e_i is aligned in $A_{initial}$. In Figure 1, $p($需要$|needs) > p($出发$|needs)$, so 需要-needs is max-probable for "needs". The definition of "max-probable from f_j to e_i" is analogous, and a link is max-probable (nondirectionally) if it is max-probable in either direction. The value of **highestLexProbRank** is the total number of max-probable links. The conditional lexical probabilities $p(e_i|f_j)$ and $p(f_j|e_i)$ are estimated using frequencies of aligned word pairs in the high-precision GIZA++ *intersection* alignments for the training corpus.

2.2.4 History Features

In addition to the above syntactic, structural, and lexical features of A, we also incorporate two features of the link deletion history itself into $Score(A)$:

linksDeleted: Total number of links deleted $A_{initial}$ thus far. At each iteration, either a link or a branch of links is deleted.

[3]While using a dynamic programming algorithm would likely improve search efficiency and allow link deletion to find an optimal solution, in practice, the greedy search runs quickly and improves alignment quality.

[4](Deng and Byrne, 2005) observe that, in a manually aligned Chinese-English corpus, 82% of the Chinese words that are aligned to multiple English words are aligned to a *contiguous* block of English words; similarly, 88% of the English words that are aligned to multiple Chinese words are aligned to a *contiguous* block of Chinese words. Thus, if a Chinese word is correctly aligned to multiple English words, those English words are likely to be "neighbors" of each other, and if an English word is correctly aligned to multiple Chinese words, those Chinese words are likely to be "neighbors" of each other.

stepsTaken: Total number of iterations thus far in the search; at each iteration, either a link or a branch is deleted. This feature serves as a constant cost function per step taken during link deletion.

2.3 Constraints

Protecting Refined Links from Deletion: Since GIZA++ refined links have higher precision than union links[5], we do not consider any GIZA++ refined links for deletion.[6]

Stoplist: In our Chinese-English corpora, the 10 most common English words (excluding punctuation marks) include {a,in,to,of,and,the}, while the 10 most common Chinese words include {了,是,在,和,的}. Of these, {a,the} and {了,的} have no explicit translational equivalent in the other language. These words are aligned with each other frequently (and erroneously) by GIZA++ union, but rarely in the gold standard. We delete all links in the set {a, an, the} $\times$ {的, 了} from $A_{initial}$ as a preprocessing step.[7]

2.4 Perceptron Training

We set the feature weights λ using a modified version of averaged perceptron learning with structured outputs (Collins, 2002). Following (Moore, 2005), we initialize the value of our expected most informative feature (**ruleCount**) to 1.0, and initialize all other feature weights to 0. During each pass over the discriminative training set, we "decode" each sentence pair by greedily deleting links from $A_{initial}$ in order to maximize the score of the resulting alignment using the current settings of λ (for details, refer to section 2.1).

We construct a set of candidate alignments $A_{candidates}$ for use in reranking as follows. Starting with $A = A_{initial}$, we iteratively explore all alignments A' in the *neighborhood* of A, adding each *neighbor* to $A_{candidates}$, then selecting the *neighbor* that maximizes $Score(A')$. When it is no longer possible to increase $Score(A)$ by deleting any links, link deletion concludes and returns the highest-scoring alignment, $A_{1\text{-}best}$.

In general, $A_{gold} \notin A_{candidates}$; following (Collins, 2000) and (Charniak and Johnson, 2005) for parse reranking and (Liang et al., 2006) for translation reranking, we define A_{oracle} as alignment in $A_{candidates}$ that is most *similar* to A_{gold}.[8] We update each feature weight λ_i as follows: $\lambda_i = \lambda_i + h_i^{A_{oracle}} - h_i^{A_{1\text{-}best}}$.[9]

Following (Moore, 2005), after each training pass, we average all the feature weight vectors seen during the pass, and decode the discriminative training set using the vector of averaged feature weights. When alignment quality stops increasing on the discriminative training set, perceptron training ends.[10] The weight vector returned by perceptron training is the average over the training set of all weight vectors seen during all iterations; averaging reduces overfitting on the training set (Collins, 2002).

3 Experimental Setup

3.1 Data Sets

We evaluate the effect of link deletion upon alignment quality and translation quality for two Chinese-English data sets, and one Arabic-English data set. Each data set consists of newswire, and contains a small subset of manually aligned sentence pairs. We divide the manually aligned subset into a training set (used to discriminatively set the feature weights for link deletion) and a test set (used to evaluate the impact of link deletion upon alignment quality). Table 1 lists the source and the size of the manually aligned training and test sets used for each alignment task.

[5] On a 400-sentence-pair Chinese-English data set, GIZA++ union alignments have a precision of 77.32 while GIZA++ refined alignments have a precision of 85.26.

[6] To see how GIZA++ refined alignments compare to GIZA++ union alignments for syntax-based translation, we compare systems trained on each set of alignments for Chinese-English translation task A. Union alignments result in a test set BLEU score of 41.17, as compared to only 36.99 for refined.

[7] The impact upon alignment f-measure of deleting these stoplist links is small; on Chinese-English Data Set A, the f-measure of the baseline GIZA++ union alignments on the test set increases from 63.44 to 63.81 after deleting stoplist links, while the remaining increase in f-measure from 63.81 to 75.14 (shown in Table 3) is due to the link deletion algorithm itself.

[8] We discuss alignment similarity metrics in detail in Section 3.2.

[9] (Liang et al., 2006) report that, for translation reranking, such *local* updates (towards the oracle) outperform *bold* updates (towards the gold standard).

[10] We discuss alignment quality metrics in detail in Section 3.2.

Using the feature weights learned on the manually aligned training set, we then apply link deletion to the remainder (non-manually aligned) of each bilingual data set, and train a full syntax-based statistical MT system on these sentence pairs. After maximum BLEU tuning (Och, 2003a) on a held-out tuning set, we evaluate translation quality on a held-out test set. Table 2 lists the source and the size of the training, tuning, and test sets used for each translation task.

3.2 Evaluation Metrics

AER (Alignment Error Rate) (Och and Ney, 2003) is the most widely used metric of alignment quality, but requires gold-standard alignments labelled with "sure/possible" annotations to compute; lacking such annotations, we can compute alignment f-measure instead.

However, (Fraser and Marcu, 2007a) show that, in phrase-based translation, improvements in AER or f-measure do not necessarily correlate with improvements in BLEU score. They propose two modifications to f-measure: varying the precision/recall tradeoff, and *fully-connecting* the alignment links before computing f-measure.[11]

Weighted Fully-Connected F-Measure Given a hypothesized set of alignment links H and a gold-standard set of alignment links G, we define $H^+ = fullyConnect(H)$ and $G^+ = fullyConnect(G)$, and then compute:

$$f\text{-}measure(H^+) = \frac{1}{\frac{\alpha}{precision(H^+)} + \frac{1-\alpha}{recall(H^+)}}$$

For phrase-based Chinese-English and Arabic-English translation tasks, (Fraser and Marcu, 2007a) obtain the closest correlation between weighted fully-connected alignment f-measure and BLEU score using α=0.5 and α=0.1, respectively. We use weighted fully-connected alignment f-measure as the training criterion for link deletion, and to evaluate alignment quality on training and test sets.

Rule F-Measure To evaluate the impact of link deletion upon rule quality, we compare the rule precision, recall, and f-measure of the rule set extracted from our hypothesized alignments and a Collins-style parser against the rule set extracted from gold alignments and gold parses.

[11] In Figure 1, the fully-connected version of the alignments shown would include the links 需要-starts and 需要- out.

Language	Train	Test
Chinese-English A	400	400
Chinese-English B	1500	1500
Arabic-English	1500	1500

Table 1: Size (sentence pairs) of data sets used in alignment link deletion tasks

BLEU For all translation tasks, we report case-insensitive NIST BLEU scores (Papineni et al., 2002) using 4 references per sentence.

3.3 Experiments

Starting with GIZA++ union (IBM Model 4) alignments, we use perceptron training to set the weights of each feature used in link deletion in order to optimize weighted fully-connected alignment f-measure (α=0.5 for Chinese-English and α=0.1 for Arabic-English) on a manually aligned discriminative training set. We report the (fully-connected) precision, recall, and weighted alignment f-measure on a held-out test set after running perceptron training, relative to the baseline GIZA++ union alignments. Using the learned feature weights, we then perform link deletion over the GIZA++ union alignments for the entire training corpus for each translation task. Using these alignments, which we refer to as "GIZA++ union + link deletion", we train a syntax-based translation system similar to that described in (Galley et al., 2006). After extracting string-to-tree translation rules from the aligned, parsed training corpus, the system assigns weights to each rule via frequency estimation with smoothing. The rule probabilities, as well as trigram language model probabilities and a handful of additional features of each rule, are used as features during decoding. The feature weights are tuned using minimum error rate training (Och and Ney, 2003) to optimize BLEU score on a held-out development set. We then compare the BLEU score of this system against a baseline system trained using GIZA++ union alignments.

To determine which value of α is most effective as a training criterion for link deletion, we set α=0.4 (favoring recall), 0.5, and 0.6 (favoring precision),

Language	Train	Tune	Test1	Test2
Chinese-English A	9.8M/newswire	25.9k/NIST02	29.0k/NIST03	–
Chinese-English B	12.3M/newswire	42.9k/newswire	42.1k/newswire	–
Arabic-English	174.8M/newswire	35.8k/NIST04-05	40.3k/NIST04-05	53.0k/newswire

Table 2: Size (English words) and source of data sets used in translation tasks

and compare the effect on translation quality for Chinese-English data set A.

4 Results

For each translation task, link deletion improves translation quality relative to a GIZA++ union baseline. For each alignment task, link deletion tends to improve fully-connected alignment precision more than it decreases fully-connected alignment recall, increasing weighted fully-connected alignment f-measure overall.

4.1 Chinese-English

On Chinese-English translation task A, link deletion increases BLEU score by 1.26 points on tuning and 0.76 points on test (Table 3); on Chinese-English translation task B, link deletion increases BLEU score by 1.38 points on tuning and 0.49 points on test (Table 3).

4.2 Arabic-English

On the Arabic-English translation task, link deletion improves BLEU score by 0.84 points on tuning, 0.18 points on test1, and 0.56 points on test2 (Table 3). Note that the training criterion for Arabic-English link deletion uses α=0.1; because this penalizes a loss in recall more heavily than it rewards an increase in precision, it is more difficult to increase weighted fully-connected alignment f-measure using link deletion for Arabic-English than for Chinese-English. This difference is reflected in the average number of links deleted per sentence: 4.19 for Chinese-English B (Table 3), but only 1.35 for Arabic-English (Table 3). Despite this difference, link deletion improves translation results for Arabic-English as well.

4.3 Varying α

On Chinese-English data set A, we explore the effect of varying α in the weighted fully-connected

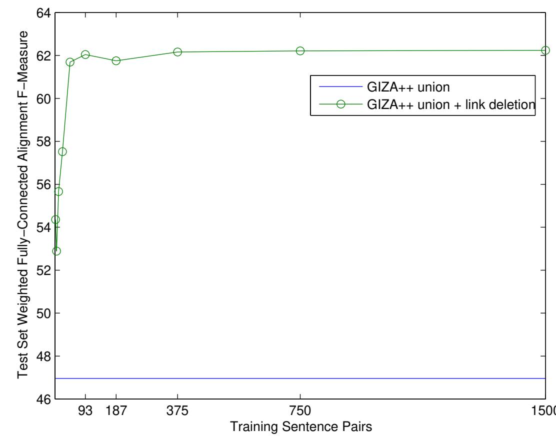

Figure 2: Effect of discriminative training set size on link deletion accuracy for Chinese-English B, α=0.5

alignment f-measure used as the training criterion for link deletion. Using α=0.5 leads to a higher gain in BLEU score on the test set relative to the baseline (+0.76 points) than either α=0.4 (+0.70 points) or α=0.6 (+0.67 points).

4.4 Size of Discriminative Training Set

To examine how many manually aligned sentence pairs are required to set the feature weights reliably, we vary the size of the discriminative training set from 2-1500 sentence pairs while holding test set size constant at 1500 sentence pairs; run perceptron training; and record the resulting weighted fully-connected alignment f-measure on the test set. Figure 2 illustrates that using 100-200 manually aligned sentence pairs of training data is sufficient for Chinese-English; a similarly-sized training set is also sufficient for Arabic-English.

4.5 Effect of Link Deletion on Extracted Rules

Link deletion increases the *size* of the extracted grammar. To determine how the *quality* of the extracted grammar changes, we compute the rule pre-

Language	Alignment	Prec	Rec	α	F-measure	Links Del/ Sent	Grammar Size	BLEU Tune	Test1	Test2
Chi-Eng A	GIZA++ union	54.76	**75.38**	0.5	63.44	–	23.4M	41.80	41.17	–
Chi-Eng A	GIZA++ union + link deletion	**79.59**	71.16	0.5	**75.14**	4.77	59.7M	**43.06**	**41.93**	–
Chi-Eng B	GIZA++ union	36.61	**66.28**	0.5	47.16	–	28.9M	39.59	41.39	–
Chi-Eng B	GIZA++ union + link deletion	**65.52**	59.28	0.5	**62.24**	4.19	73.0M	**40.97**	**41.88**	–
Ara-Eng	GIZA++ union	35.34	**84.05**	0.1	73.87	–	52.4M	54.73	50.9	38.16
Ara-Eng	GIZA++ union + link deletion	**52.68**	79.75	0.1	**75.85**	1.35	64.9M	**55.57**	**51.08**	**38.72**

Table 3: Results of link deletion. Weighted fully-connected alignment f-measure is computed on alignment test sets (Table 1); BLEU score is computed on translation test sets (Table 2).

Alignment	Parse	Rule Precision	Recall	F-measure	Total Non-Unique
gold	gold	100.00	100.00	100.00	12,809
giza++ union	collins	**50.49**	44.23	47.15	11,021
giza++ union+link deletion, α=0.5	collins	47.51	53.20	**50.20**	13,987
giza++ refined	collins	44.20	**54.06**	48.64	15,182

Table 4: Rule precision, recall, and f-measure of rules extracted from 400 sentence pairs of Chinese-English data

cision, recall, and f-measure of the GIZA++ union alignments and various link deletion alignments on a held-out Chinese-English test set of 400 sentence pairs. Table 4 indicates the total (non-unique) number of rules extracted for each alignment/parse pairing, as well as the rule precision, recall, and f-measure of each pair. As more links are deleted, more rules are extracted–but of those, some are of good quality and others are of bad quality. Link-deleted alignments produce rule sets with higher rule f-measure than either GIZA++ union or GIZA++ refined.

5 Conclusion

We have presented a link deletion algorithm that improves the precision of GIZA++ union alignments without notably decreasing recall. In addition to lexical and structural features, we use features of the extracted syntax-based translation rules. Our method improves alignment quality and translation quality on Chinese-English and Arabic-English translation tasks, relative to a GIZA++ union baseline. The algorithm runs quickly, and is easily applicable to other language pairs with limited amounts (100-200 sentence pairs) of manually aligned data available.

Acknowledgments

We thank Steven DeNeefe and Wei Wang for assistance with experiments, and Alexander Fraser and Liang Huang for helpful discussions. This research was supported by DARPA (contract HR0011-06-C-0022) and by a fellowship from AT&T Labs.

References

Peter F. Brown, Stephen Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. *The Mathematics of Statistical Machine Translation: Parameter Estimation*. Computational Linguistics, Vol. 19, No. 2, 1993.

Eugene Charniak and Mark Johnson. *Coarse-to-fine n-best parsing and MaxEnt discriminative reranking*. Proceedings of ACL, 2005.

Colin Cherry and Dekang Lin. *Soft Syntactic Constraints for Word Alignment through Discriminative Training*. Proceedings of ACL (Poster), 2006.

David Chiang. *A Hierarchical Phrase-Based Model for Statistical Machine Translation*. Proceedings of ACL, 2005.

David Chiang. *Hierarchical phrase-based translation*. Computational Linguistics, 2007.

Michael Collins. *Discriminative Reranking for Natural Language Parsing*. Proceedings of ICML, 2000.

Michael Collins. *Discriminative training methods for hidden Markov models: theory and experiments with perceptron algorithms*. Proceedings of EMNLP, 2002.

John DeNero and Dan Klein. *Tailoring Word Alignments to Syntactic Machine Translation*. Proceedings of ACL, 2007.

Yonggang Deng and William Byrne. *HMM word and phrase alignment for statistical machine translation*. Proceedings of HLT/EMNLP, 2005.

Alexander Fraser and Daniel Marcu. *Measuring Word Alignment Quality for Statistical Machine Translation*. Computational Linguistics, Vol. 33, No. 3, 2007.

Alexander Fraser and Daniel Marcu. *Getting the Structure Right for Word Alignment: LEAF*. Proceedings of EMNLP, 2007.

Michel Galley, Mark Hopkins, Kevin Knight, and Daniel Marcu. *What's in a Translation Rule?* Proceedings of HLT/NAACL-04, 2004.

Michel Galley, Jonathan Graehl, Kevin Knight, Daniel Marcu, Steve DeNeefe, Wei Wang, and Ignacio Thayer. *Scalable Inference and Training of Context-Rich Syntactic Translation Models*. Proceedings of ACL, 2006.

Liang Huang, Kevin Knight, and Aravind Joshi. *Statistical Syntax-Directed Translation with Extended Domain of Locality*. Proceedings of AMTA, 2006.

Abraham Ittycheriah and Salim Roukos. *A Maximum Entropy Word Aligner for Arabic-English Machine Translation*. Proceedings of HLT/EMNLP, 2005.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. *Statistical Phrase-Based Translation*. Proceedings of HLT/NAACL, 2003.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, Evan Herbst. *Moses: Open Source Toolkit for Statistical Machine Translation*. Proceedings of ACL (demo), 2007.

Percy Liang, Alexandre Bouchard-Cote, Dan Klein, and Ben Taskar. *An end-to-end discriminative approach to machine translation*. Proceedings of COLING/ACL, 2006.

Yang Liu, Qun Liu, and Shouxun Lin. *Log-linear Models for Word Alignment*. Proceedings of ACL, 2005.

Yang Liu, Qun Liu, and Shouxun Lin. *Tree-to-String Alignment Template for Statistical Machine Translation*. Proceedings of ACL, 2006.

Adam Lopez and Philip Resnik. *Improved HMM Alignment Models for Languages with Scarce Resources*. Proceedings of the ACL Workshop on Parallel Text, 2005.

Jonathan May and Kevin Knight. *Syntactic Re-Alignment Models for Machine Translation*. Proceedings of EMNLP-CoNLL, 2007.

Robert C. Moore. *A Discriminative Framework for Bilingual Word Alignment*. Proceedings of HLT/EMNLP, 2005.

Robert C. Moore, Wen-tau Yih, and Andreas Bode. *Improved discriminative bilingual word alignment*. Proceedings of ACL, 2006.

Franz Josef Och. *Minimum Error Rate Training in Statistical Machine Translation*. Proceedings of ACL, 2003.

Franz Josef Och and Hermann Ney. *A Systematic Comparison of Various Statistical Alignment Models*. Computational Linguistics, Vol. 29, No. 1, 2003.

Franz Josef Och and Hermann Ney. *The alignment template approach to statistical machine translation*. Computational Linguistics, 2004.

K. Papineni, S. Roukos, T. Ward, and W. J. Zhu. *BLEU: a Method for Automatic Evaluation of Machine Translation*. Proceedings of ACL, 2002.

Chris Quirk, Arul Menezes, and Colin Cherry. *Dependency Treelet Translation: Syntactically Informed Phrasal SMT*. Proceedings of ACL, 2005.

Ben Taskar, Simon Lacoste-Julien, and Dan Klein. *A Discriminative Matching Approach to Word Alignment*. Proceedings of HTL/EMNLP, 2005.

Stephan Vogel, Hermann Ney, and Christoph Tillmann. *HMM-Based Word Alignment in Statistical Translation* Proceedings of COLING, 1996.