Improving MT Word Alignment Using Aligned Multi-Stage Parses

Adam Meyers[†], **Michiko Kosaka**[‡], **Shasha Liao**[†] **and Nianwen Xue**[◇]
[†] New York University, [‡]Monmouth University, [◇]Brandeis University

Abstract

We use hand-coded rules and graph-aligned logical dependencies to reorder English text towards Chinese word order. We obtain a 1.5% higher F-score for Giza++ compared to running with unprocessed text. We describe this research and its implications for SMT.

1 Introduction

Some statistical machine translation (SMT) systems use pattern-based rules acquired from linguistically processed bitexts. They acquire these rules through the alignment of a parsed structure in one language with a raw string in the other language (Yamada and Knight, 2001; Shen et al., 2008) or the alignment of source/target language parse trees (Zhang et al., 2008; Cowan, 2008). This paper shows that machine translation (MT) can also benefit by aligning a "deeper" level of analysis than parsed text, which includes semantic role labeling, regularization of passives and wh constructions, etc. We create GLARF representations (Meyers et al., 2009) for English and Chinese sentences, in the form of directed acyclic graphs. We describe two graph-based techniques for reordering English sentences to be closer to that of corresponding Chinese sentences. One technique is based on manually created rules and the other is based on an automatic alignment of GLARF representations of Chinese/English sentences. After reordering, we align words of the reordered English with the words of the Chinese, using the Giza++ word aligner(Och and Ney, 2003). For both techniques, the resulting alignment has a higher F-score than Giza++ on raw text (a 0.7% to 1.5% absolute improvement). In principle, our reordered text can be used to improve any Chinese/English SMT system for which Giza++ (or other word aligners) are part of the processing pipeline.

These experiments are a first step in using GLARF-style analyses for MT, potentially improving systems that already perform well with aligned text lacking large gaps in surface alignment. We hypothesize that SMT systems are most likely to benefit from deep analysis for structures where source and target language word order differs the most. We propose using deep analysis to reorder such structures in one language to more closely reflect the word order of the other language. The text would be reordered at two stages in an SMT system: (1) prior to acquiring a translation model; and (2) either prior to translation (if source text is reordered) or after translation (if target text is reordered). Our system moves large constituents (e.g., noun post-modifiers) to bring English word order closer to that of parallel Chinese sentences. This improves word alignment and is likely to improve SMT.

For this work we use two English/Chinese bitext corpora developed by the Linguistic Data Consortium (LDC): the Tides FBIS corpus and the GALE Y1 Q4 Chinese/English Word-Alignment corpus. We used 2300 aligned sentences from FBIS for development purposes. We divided the GALE corpus into into a 3407 sentence development subcorpus (DEV) and a 1505 sentence test subcorpus (TEST). We used the LDC's manual alignments of the FBIS corpus to score these data.

Proceedings of SSST-5, Fifth Workshop on Syntax, Semantics and Structure in Statistical Translation, pages 88–97,
ACL HLT 2011, Portland, Oregon, USA, June 2011.

2 Related Work in SMT

Four papers stand out as closely related to the present study. (Collins et al., 2005; Wang et al., 2007) describe experiments which use manually created parse-tree-based rules to reorder one side of a bitext: German/English in (Collins et al., 2005) and English/Chinese in (Wang et al., 2007). Both achieve BLEU score improvements for SMT: 25.2% to 26.8% for (Collins et al., 2005) and 28.52 to 30.86 for (Wang et al., 2007). (Wang et al., 2007) uses rules very similar to our own as they use the same language pair, although they reorder the Chinese, whereas we reorder the English. The most significant differences between our research and (Collins et al., 2005; Wang et al., 2007) are: (1) our manual rules benefit from a level of representation "deeper" than a surface parse; and (2) In addition to the hand-coded rules, we also use automatic alignment-based rules. (Wu and Fung, 2009) uses PropBank role labels (Palmer et al., 2005) as the basis of a second pass filter over an SMT system to improve the BLEU score from 42.99 to 43.51. The main similarity to the current study is the use of a level of representation that is "deeper" than a surface parse. However, our application of linguistic structure is more like that of (Wang et al., 2007) and our "deep" level connects all predicates and arguments in the sentence, regardless of part of speech, rather than just connecting verbs to their arguments. (Bryl and van Genabith, 2010) describes an open source LFG F-structure alignment tool with an algorithm similar to our previous work. They evaluate their alignment output on 20 manually-aligned German and English F-structures. They leave the impact of their work on MT to future research.

In addition to these papers, there has also been some work on rule-based reordering preprocessors to word alignment based on shallower linguistic information. For example (Crego and Mariño, 2006) reorders based on patterns of POS tags. We hypothesize that this is similar to the above approaches in that patterns of POS tags are likely to simulate parsing or chunking.

3 Preparing the Data

The two stage parsers of previous decades (Hobbs and Grishman, 1976) generated a syntactic representation analogous to the (more accurate) output of current treebank-based parsers (Charniak, 2001) and an additional second stage output that regularized constructions (passive, active, relative clauses) to representations similar to active clauses with no gaps, e.g., *The book was read by Mary* was given a representation similar to that of *Mary read the book*. Treating the active clause as canonical provides a way to reduce variation in language and thus, making it easier to acquire and apply statistical information from corpora–there is more evidence for particular statistical patterns when applications learn patterns and patterns more readily match data.

Two-stage parsers were influenced by linguistic theories (Harris, 1968; Chomsky, 1957; Bresnan and Kaplan, 1982) which distinguish a "surface" and a "deep" level. The deep level neutralizes differences between ways to express the same meaning–a passive like *The cheese was eaten by rats* was analyzed in terms of the active form *Rats ate the cheese*. Currently "semantic parsing" refers to a similar representation, e.g., (Wagner et al., 2007) or our own GLARF (Meyers et al., 2009). However, the term is also used for semantic role labelers (Gildea and Jurafsky, 2002; Xue, 2008), systems which typically label semantic relations between verbs and their arguments and rarely cover arguments of other parts of speech. Second stage semantic parsers like our own, connect all the tokens in the sentence. Aligned text processed in this way can (for example) represent differences in English/Chinese noun modifier order, including relative clauses. In contrast, few role labelers handle noun modifiers and none handle relative clauses. Below, we describe the GLARF framework and our system for generating GLARF representations of English and Chinese sentences.

For each language, we combine several types of information which may include: named entity (NE) tagging, date/number regularization, recognition of multi-word expressions (the preposition *with respect to*, the noun *hand me down* and the verb *ad lib*), role labels for predicates of all parts of speech, regularizing passives and other constructions, error correction, among other processes into a single typed feature structure (TFS) representation. This TFS is converted into a set of 25-tuples representing dependency-style relations between pairs of words in the sentence. Three types of dependencies are

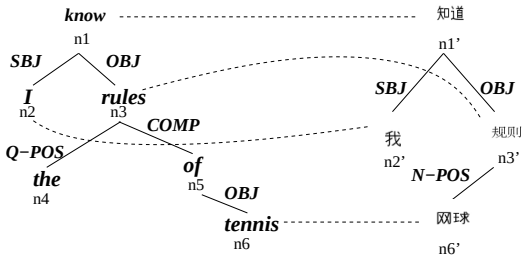

Figure 1: Word-Aligned Logic1 Dependencies

represented: *surface* dependencies (close to the level of the parser), *logic1* dependencies (reflecting various regularizations) and *logic2* dependencies (reflecting the output of a PropBanker, NomBanker and Penn Discourse Treebank transducer).(Palmer et al., 2005; Xue and Palmer, 2003; Meyers et al., 2004; Miltsakaki et al., 2004) The surface dependency graph is a tree; The logic1 dependency graph is an directed acyclic graph; and The logic2 dependency graph is a directed graph with cycles, covering only a subset of the tokens in the sentence. For these experiments, we focus on the logic1 relations, but will sometimes use the surface relations as well. Figure 1 is a simple dependency-based logic1 representation of *I know the rules of tennis* and its Chinese translation. The edge labels name the relations between heads and dependents, e.g., *I* is the SBJ of *know* and the dashed lines indicate word level correspondences. Each node is labeled with both a word and a unique node identifier (n1, n1', etc.)

The English system achieves F-scores for logic1 dependencies on parsed news text in the 80–90% range and the Chinese system achieves F-scores in the 74–84% range, depending on the complexity of the text. The English system has been created over the course of about 9 years, and consequently is more extensive than the Chinese system, which has been created over the past 3 years. The systems are described in more detail in (Meyers et al., 2009).

The GLARF representations are created in a series of steps involving several processors. The English pipeline includes: (1) dividing text into sentences; (2) running the JET NE tagger (Ji and Grishman, 2006); (3) running scripts that clean up data (to prevent parser crashes); (4) running a parser (currently Charniak's 2005 parser based on (Charniak, 2001)); (5) running filters that: (a) correct common parsing errors; (b) merge NE information with the parse, resolving conflicts in constituent boundaries by hand-coded rules; (c) regularize numbers, dates, times and holidays; (d) identify heads and label relations between constituents; (e) regularize text grammatically (filling empty subjects, resolving relative clause and Wh gaps, etc.); (f) mark conjunction scope; (g) identify transparent constituents (e.g., recognizing, that *A variety of different people* has the semantic features of *people* (human), not those of *variety*, the syntactic head of the phrase.); among other aspects. The Chinese pipeline is similar, except that it includes the LDC word segmenter and a PropBanker (Xue, 2008). Also, the regularization routines are not as completely developed, e.g., relative clause gaps and passives are not handled yet. The Chinese system currently uses the Berkeley parser (Petrov and Klein, 2007). Each of these pipelines derives typed feature structure representations, which are then converted into the 25 tuple representation of 3 types of dependencies between pairs of tokens: surface, logic1 and logic2.

To insure that the logic1 graphs are acyclic, we assume that certain edges are surface only and that the resulting directed acyclic graphs can have multiple roots. It turns out that the multiple rooted cases are mostly limited to a few constructions, the most common being parenthetical clauses and relative clauses. A parenthetical clause takes the main clause as an argument. For example, in *The word 'potato', he claimed, is spelled with a final 'e'.*, the verb *claimed*, takes the entire main clause as an argument, we assume that *he claimed* is a dependent on the main verb *(is) spelled* labeled PARENTHETICAL in our surface dependency structure, but that the main verb *(is) spelled* is a dependent of the verb *claimed* in our logic1 structure, labeled COMPLEMENT. Thus the logic1 surface dependency structure have distinct roots. In a relative clause, such as *the book that I read*", we assume that the clause *that I read* is a dependent on the noun *book* in our surface dependency structure with the label RELATIVE, but *book* is a dependent on the verb *read* in our logic1 dependency structure, with the label OBJ. This, means that our logic1 dependency graphs for sentences containing relative clauses are multi-rooted. One of the roots is the same as the root of the surface tree and the other root is the root of the relative clause graph (a rela-

tive pronoun or a main verb). Furthermore, there is a surface path connecting the relative clause root to the rest of the graph. Noncyclic graph traversal is possible, provide that: (1) we use the surface path to enter the graph representing the relative clause – otherwise, the traversal would skip the relative clause; and (2) we halt the traversal if we reach this path a second time – this avoids traversing down an endless path. The parenthetical and relative clause are representative of the handful of cases in which naive representations would introduce loops. All cases of which we are aware have the essential properties of one of these two cases: (1) either introducing a different single root of the clause; or (2) introducing an additional root that can be bridged by a surface path.

4 Manual Reordering Rules

We derived manual rules for making the English Word Order more like the Chinese by manually inspecting the data. We inspected the first 100-200 sentences of the DEV corpus by first transliterating the Chinese into English – replaced each Chinese word with the aligned English counterpart. Several patterns emerged which were easy to formalize into rules in the GLARF framework. These patterns were verified and sometimes generalized through discussions with native Chinese speakers and linguists. Our rules, similar to those of (Wang et al., 2007) are as follows (results are discussed in section 6): (1) Front a post-nominal PP headed by a preposition in the list {*of, in, with, about*)}. (2) Front post-nominal relative clause that begins with *that* or does not have any relative pronoun, such that the main predicate is not a copula plus adjective construction. (3) Front post-nominal relative clause that begins with *that* or has no relative pronoun if the main predicate is a copula+adjective construction which is not negated by a word from the set {*no neither nor never not n't*}. (4) Front post-nominal reduced relative in the form of a passive or adjectival phrase. (5) Move adverbials *more than* and *less than* after numbers that they modify. (6) Move PPs that post-modify adjectives to the position before the adjective. (7) Move subordinate conjunctions *before* and *after* to the end of the clause that they introduce. (8) Move an initial one-word-long title (*Mr., Ms., Dr., President*) to the end of the name. (9) Move temporal adverbials (adverb, PP, subordinate clause that is semantically temporal) to pre-verb position.

5 Automatic Node Alignment and its Application for Word Alignment

In this experiment, we automatically derive reorderings of the English sentences from an alignment between nodes in logic1 dependency graphs for the English (source) and Chinese (target) sentences. Source/Target designations are for convenience, since the direction of MT is irrelevant.

We define an alignment as a partial function from the nodes in the source graph and the nodes in the target graph. We, furthermore, assume that this mapping is 1 to 1 for most node pairs, but can be n to 1 (or 1 to n). Furthermore, we allow some nodes, in effect, to represent multiple tokens. These are identified as part of the GLARF analysis of a particular sentence string and reflect language-specific rules. Thus, for our purposes, a mapping between a source and target node, each representing a multi-word expression is 1 to 1, rather than N to N.

We identify the following types of multi-word expressions for this purpose: (a) idiomatic expressions from our monolingual lexicons, (b) dates, (c) times (d) numbers and (e) ACE (Grishman, 2000) NEs. Dates, holidays and times are regularized using ISO-TimeML, e.g., January 3, 1977 becomes 1977-03-01 and numbers are converted to Arabic numbers.

5.1 ALIGN-ALG1

This work uses a modified version of ALIGN-ALG1, a graph alignment algorithm we previously used to align 1990s-style two-stage parser output for MT experiments. ALIGN-ALG1 is an $O(n^2)$ algorithm, n is the maximum number of nodes in the source and target graphs (Meyers et al., 1996; Meyers et al., 1998). Given Source Tree T and Target Tree T', an $alignment(T, T')$ is a partial function from nodes N in T to nodes N' in T'. An exhaustive search of possible alignments would consider all non-intersecting combinations of the $T \times T'$ pairs of source/target nodes – There are at most $T!$ such pairings where $T >= T'$.[1] However, ALIGN-ALG1 assumes that some of these pairings are unlikely, and

[1] This ignores N to 1 matches, which we allow, although relatively rarely.

favors pairings that assume the structure of the trees correspond more closely. In particular, it is assumed that ancestor nodes are more likely to match if most of their descendant nodes match as well.

ALIGN-ALG1 finds the highest scoring alignment, where the score of an alignment is the sum of the scores of the node pairs in the partial function. The score for each node pair (n, n') partially depends on the scores of a mapping from the children of n to the children of n'. While the process of calculating the scores is recursive, it can be made efficient using dynamic programming.

ALIGN-ALG1 assumes that we align r and r', the roots of T and T'. Calculating the scores for r and r', entails calculating the scores of pairs of their children, and by extension all mappings from N to N' that obey the dominance preserving constraint: Given nodes n_1 and n_2 in N and nodes n_1' and n_2' in N', where all 4 nodes are part of the alignment, it cannot be the case that: n_1 dominates n_2, but n_1' does not dominate n_2'. Here, *dominates* means *is an ancestor in the dependency graph*. ALIGN-ALG1 scores each pair of nodes using the formula: $Score(n, n') = Lex(n, n') + ChildVal(n, n')$, where $Lex(n, n')$ is a score based on matching the words labeling nodes n and n', e.g., the score is 1 if the pair is found in a bilingual dictionary and 0 otherwise. Given n has children $c_0, \ldots, c_i$ and n' has children $c_0', \ldots, c_j'$, to calculate ChildVal: (1) Create Child-Matrix, a $(i + 1) \times (j + 1)$ matrix (2) Fill every position ($1 <= x <= i$, $1 <= x' <= j$) with $Score(x, x')$ (3) Fill every position (i+1, $1 <= x' <= j$) with $Score(n, x')$ minus a penalty (e.g., - .1) for *collapsing an edge*. This treats n' and x' as a single unit, matched to n.[2] (4) Fill every position ($1 <= x <= i$, j+1) with $Score(x, n')$ minus a penalty for *collapsing an edge*. Thus $n + x$ is paired with n'. (5) Set (i+1,j+1) to $-\infty$. Collapsing both source and target edges is not permitted. (6) For all sets of positions in the matrix such that no node or column is repeated, select the set with the highest aggregate score. The aggregate score is the numeric value of $ChildVal(n, n')$. If (n,n') is part of the alignment that is ultimately chosen, this choice of node pairs is also part of the alignment. There are at most $max(i + 1, j + 1)!$ possible pairings. Rather than calculating them all, a greedy heuristic can reduce the calculation time with minimal effect on accuracy: the highest scoring cell in the matrix is chosen first, conflicting cells are eliminated, the next highest scoring cell is chosen, etc.

Consider the example in Figure 1, assuming the dashed lines connect lexical matches (the function LEX returns 1 for these node pairs). Where n1 and n1' are the roots, $Score(n1, n1') = 1 + ChildVal(n1, n1')$. Calculating $ChildVal(n1, n1')$ requires a recursive descent down the pairs of nodes, until the bottom most pair is scored. $Score(n6, n6') = 1$. $Score(n5, n6') = 0 + .9$ (derived by collapsing an edge and subtracting a penalty of .1). $Score(n3, n3') = 1 + .9 = 1.9$. $Score(n2, n2') = 1$. $ChildVal(n1, n1') = 1 + 1.9 = 2.9$. Thus $Score(n1, n1') = 3.9$. The alignment includes: $(n1, n1')$, $(n2, n2')$, $(n3, n3')$, $(n5, n6')$, $(n6, n6')$.

The collapsing of edges helps recognize cases where multiple predicates form substructures, e.g., *take a walk, is angry*, etc. in one tree can map to single verbs in the other tree, allowing outgoing edges from *walk* or *angry* to map to outgoing edges of the corresponding verb, e.g., the agent and goal of *John walked to the store* could map to the agent and goal of *John took a walk to the store*.

In practice, ALIGN-ALG1 falls short because: (1) Our translation dictionary does not have sufficient coverage for the algorithm to perform well; (2) The assumption that the roots of both graphs should be aligned is often false. Parallel text often reflects a dynamic, rather than a literal translation. In one pair of aligned sentences in the FBIS corpus, the English phrase *the above mentioned requests* corresponds to: 陈水扁的这些要求 meaning *these requests of Chen Shui-bian – Chen Shui-bian* has no counterpart in the English. Parts of translations can be omitted due to: (a) the discretion of the translators, (b) the expected world knowledge of particular language communities, (c) the cultural importance of particular information, etc.; (3) Violations of the dominance-preserving constraint exist. The most common type that we have observed consists of sequences of transparent nouns and *of* (e.g., *series of*) in English corresponding to quantifiers in

[2]The slight penalty represents that collapsing edges complicate the analysis and is thus disfavored (Occam's Razor).

Chinese (一系列). Thus the head of the English construction corresponds to the dependent of the Chinese construction and vice versa.

5.2 Lexical Resources

Our primary bilingual Chinese/English dictionary (LEX1) had insufficient coverage for ALIGN-ALG1 to be effective. LEX1 is a merger between: The LDC 2002 Chinese-English Dictionary and HowNet. In addition, we manually added additional translations of units of measure from English. We also used NEDICT, a name translation dictionary (Ji et al., 2009) and AUTODICT, English/Chinese word to word pairs with high similarity scores taken from MT phase tables created as part of the (Zhang et al., 2007) system. The NEDICT was used both for precise matches and partial matches (since, NEs can often be synonymous with substrings of NEs). In addition, we used some WordNet (Fellbaum, 1998) synonyms of English to expand the coverage of all the dictionaries, allowing English words to match Chinese word translations of their synonyms. We allowed additional matches of function words that served similar functions in the two languages including: copulas, pronouns and determiners.

Finally, we use a mutual information (MI) based approach to find further lexical information. We run our alignment program over the corpus two times, the first time, we acquire statistical information useful for generating a MI-based score. This score is used as a lexical score on the second pass for items that do not match any of the dictionaries. On the first pass, we tally the frequency of each pair of source/target words s and t, such that neither s, nor t are matched lexically to any other item in the sentence. We, furthermore, keep track of the number of times each word appears in the corpus and the number of times each word appeared unaligned in the corpus. We tally MI as follows: $\frac{pair-frequency^2}{1+(source-word-frequency \times target-word-frequency)}$ One is added to the denominator as a variation on add-one smoothing (Laplace, 1816), intended to penalize low frequency scores. We calculate this score in two ways: (a) using the global frequencies of the source and target words; and (b) using the frequency these words were unaligned. The larger of the two scores is the one that is actually use.

Different lexicons are given different weights. Matches between words in the hand-coded translation dictionary and NEDICT are given a score of 1.0. Matches in other dictionaries are allotted lower scores to represent that these are based on automatically acquired information, which we assume is less reliable than manually coded information.[3]

5.3 ALIGN-ALG2

With ALIGN-ALG2, we partially address two limitations of ALIGN-ALG1: (1) the assumption that the roots of source and target graph are aligned; and (2) the dominance-preserving constraint. Basically, we assume that structural similarity is favored, but not necessarily at the global level. Thus it is likely that many subparts of corresponding trees correspond closely, but not necessarily the highest nodes in the trees.

We use ALIGN-ALG1 to align every possible pair of S source nodes and T target nodes. Then we look for P, the highest scoring node pair of all SXT pairs. P and all the pairs of descendants that are used to derive this score (the highest scoring pairs of children, grand children, etc.) become the initial output. Then we find all unmatched source and target children, and look up the highest scoring pair of these nodes, and we repeat the process, adding the resulting node pairs to the output. We continue to repeat this process until either all the nodes are included in the output or there is no remaining pair with a score above a threshold score (we leave automatic methods of tuning this score to future work and preliminarily have set this parameter to .3). This means that: 1) some parts of the graphs are left unaligned (the alignment is a partial mapping); 2) the alignment is more resilient to misalignment caused by differences in graph structure, regardless of the reason; and 3) the alignment may be between pair of unconnected graphs, each containing subsets of nodes and edges in the source and target graphs. While more complex than ALIGN-ALG1, ALIGN-ALG2 performs relatively quickly. After one iteration using ALIGN-ALG1, scores are looked up, not recalculated.

[3]Current informal weights of .2 to .6 may be replaced with automatically tuned weights (hill-climbing, etc.) in future work.

5.4 Treating Multiple Tokens as One

In some cases, parsing and segmentation of text can be corrected through minor modifications to our alignment routine. Similarly, we use bilingual lexical information to determine that certain other adjacent tokens should be treated as single words for purposes of alignment.

Given a language for which segmentation is a common source of processing error (Chinese), if a token is unaligned, we check to see whether subdividing the token into two sub-tokens would allow one or both of these sub-tokens to be alignable with unaligned tokens in the other language. We iterate through the string one token at a time, trying all partitions. Given a source token *ABC*, consisting of segments *A*, *B* and *C*, we test the two pairs of subsequences {*A*, *BC*} and {*AB*, *C*}, to see which of the two partitions (if any) could be aligned with unaligned target tokens and we compare the scores of both, selecting the highest score. Unless no partition yields further source/target matches, we then choose the highest scoring partition and add the resulting node pairings to our alignment. In a similar way, if there are a pair of aligned names consisting of source tokens $s_j \ldots s_k$ and target tokens $t_j \ldots t_k$, we look for adjacent unaligned source nodes (a sequence of nodes ending in s_{j-1} or beginning with s_{k+1}) and/or adjacent target language nodes, such that adding these nodes to the name sequence would produce at least as high a lexical score. The lexicon can also be used to match two adjacent items to the same word. We use a similar routine that checks our lexicons for words that are adjacent to matching words. This is particularly meaningful for the entries automatically acquired by means of MI, as our current method for acquiring MI would not distinguish between 1 to 1 and N to 1 cases. Thus MI scores for adjacent items typically does mean that an N to 1 match is appropriate. For example, the Chinese word 特命全权大使 had high MI with every word in the sequence (except *and*): *ambassador extraordinary and plenipotentiary* (example is from FBIS). This routine was able to cause our procedure to treat this English sequence as a single token.

5.5 Using Node Alignment for Reordering

Given a node alignment, we can attempt to reorder the source language so that words associated with aligned nodes reflect the order of the words labeling the corresponding target nodes. Specifically, we reorder our surface phrase structure-based representation of the source language (English) and then print out all the words yielded from the resulting reordered tree. Reordering takes place in a bottom up fashion as follows: for each phrase P with children $c_0 \ldots c_n$, reorder the structure beneath the child nodes first. Then build the new-constituent right to left, one child at a time from $c_n \ldots c_0$. Starting with an empty sequence, each item is put in its proper place among the constituents in the sequence so far. At each step, place some c_i after some c_j in $c_{i+1} \ldots c_n$, such that c_j $align\_precedes$ c_i and c_j is after every c_k in $c_{i+1} \ldots c_n$ such that c_i $align\_precedes$ c_k. If c_j does not exist, c_i is placed at the beginning of the sequence so far.

Definition of X $align\_precedes$ Y, where X and Y are nodes sharing the same parent: (1) Let $pairs_X$ be the set of source/target pairs in the alignment such that some (leaf node) descendant of X is the source node in the pair; (2) Let $pairs_Y$ be the set of pairs in the alignment such that some descendant of Y is the source node in the pair; (3) let X_{tmax} be the last target member of a pair in $pairs_X$, where the order is determined by the word order of the target words labeling the nodes; (4) let Y_{tmin} be the first target member of a pair in $pairs_Y$, where the order is determined the same way; (5) let X_{smin} be the first source member of a pair in $pairs_x$, according to the source sentence word order; (6) let Y_{smax} be the last source word in a pair in $pairs_Y$ ordered the same way. (7) X $align\_precedes$ Y if: X_{tmax} precedes Y_{tmin} and there is no source/target pair Q, R in the alignment such that: (A) R precedes, Y_{tmin}; (B) X_{tmax} precedes R; (C) Q either precedes X_{smin} or follows Y_{smax}; (D) If Q precedes Y_{smax}, then R does not precede Y_{tmin}.

Essentially, the $align\_precedes$ operator provides a conservative way to order the source subtrees S_1 and S_2 by their aligned target sub-tree counterparts T_1 and T_2. The idea is that if T_1 and T_2 are ordered in an opposite manner to S_1 and S_2, the source subtrees should trade places. However,

System	DEV	TEST
BASELINE	53.1%	49.9%
MANUAL	54.0% ($p < .01$)	50.6% (not significant)
ALIGN	53.5% ($p < .05$)	51.1% ($p < .01$)
ALIGN+MI	53.8% ($p < .01$)	51.4% ($p < .01$)

Table 1: F Scores for Reordering Rules

a source/target pair B_s, B_t can block this reordering if doing so would upset the order of the moved constituents relative to B_s and B_t e.g., if before the move, B_s precedes S_2 and B_t precedes T_2, but after the move S_2 would precede B_s. This reordering proceeds from right to left, halting after placing c_0.

6 Results

The results summarized in table 1, provide F-scores (the harmonic mean of precision and recall) of the word alignment resulting from running GIZA++ with and without our reordering rules, using the LDC's manually created word alignments for our DEV and TEST corpora.[4] Giza++ is run with English as source and Chinese as target. Our baseline is the result of running Giza++ on the raw text. The statistical significance of differences from the baseline are provided in parentheses, next to each non-baseline score(rounded to 2 significant digits). We divided both corpora into 20 parts and ran all versions of the program on each section. We compared the system output for each section against the baseline and used the sign test to calculate statistical significance. All system output except one[5] achieved at least $p < .05$ and most systems achieved significance well below $p < .01$.

Informally, we observe that the rules reordering common noun modifiers produce most of the total improvement. However, space limitations prevent a detailed exploration of these differences. The results show that for both DEV and TEST corpora, both reordering approaches improve F-scores of GIZA++ over the baseline. The manual rules (MANUAL) seem to suffer somewhat from overtraining on the DEV corpus, as they were designed based on DEV corpus examples, whereas the alignment based approaches (ALIGN and subsequent entries in the table) seem resilient to these effects. The use of Mutual Information (ALIGN+MI) seems to further improve the F-score.

The two approaches worked for many of the same phenomena, e.g., they fronted many of the same noun post-modifiers. The advantage of the hand-coded rules seems to be that they cover reordering of words which we cannot align. For example, a rule that fronts post-nominal *of* phrases operates regardless of dictionary coverage. Thus the rule-based version fronted the *of* phrase in the NP *the government of the Guangxi Zhuangzu Autonomous Region* in our DEV corpus, due to the absolute application of the rule. However, the alignment-based version did not front the PP because the name was not found in NEDICT. On the other hand, exceptions to this rule were better handled by the alignment-based system. For example, if *series of* aligns with the quantifier 一系列, the PP would be incorrectly fronted by the manual, but not the alignment-based system. Also, the alignment-based method can handle cases not covered by our rules with minimal labor. Thus, the automatic system, but not the manual-rule system fronted the locative PP *in Guangxi* to the position between *been* and *quite* in the sentence: *foreign businessmen have been quite actively investing in Guangxi*. This is closer to the Chinese, but may have been difficult to predict with an automatic rule for several reasons, e.g., it is not clear if all post-verbal locative phrases should front.

We further analyzed the DEV ALIGN+MI run to determine both how often nodes were combined together by our algorithm to produce N to 1 alignments and the number of reorderings undertaken. It turns out that out of the 59,032 pairs of nodes were aligned for 3076 sentence pairs:[6] 55,391 alignments

[4] We used F-scores, which (Fraser and Marcu, 2007) show to correlate well with improvements in BLEU. We weighted precision and recall evenly since we do not currently have BLEU scores for MT that use these alignments and therefore cannot tune the weights. Our results also showed improvements in alignment error rate (AER) (Och and Ney, 2000), which incorporate the "possible" and "sure" portions of the manual alignment into F-score, but do not seem to correlate well with BLEU.

[5] When run on the test corpus, the manual system outperformed the baseline system on only 13 out of 20 sections.

[6] When sentences were misparsed in one language or the other they were not reordered by the program.

were 1 to 1 (93.8% of the total) , 3443 alignments were 2 to 1 (5.8% of the total) and 203 alignments were N to 1, where N is greater than 2 (0.3% of the total). The reordering program moved 1597 single tokens; 2140 blocks 2 or 3 tokens long; 1203 blocks of 4 or 5 tokens; 610 blocks of 6 or 7 tokens, 419 blocks of 8, 9 or 10 tokens, and 383 blocks of more than 10 tokens.

7 Concluding Remarks

We have demonstrated that deep level linguistic analysis can be used to improve word alignment results. It is natural to consider whether or not these reorderings are likely to improve MT results. Both the manual and alignment-based systems moved post-nominal English modifiers to pre-nominal position, to reflect Chinese word order – other movements were much less frequent. In principle, these selective reorderings may help SMT systems identify *phrases* of English that correspond to *phrases* of Chinese, thus improving the quality of the phrase tables, especially when large chunks are moved. We would also expect that the precision of our system to be more important than the recall, since our system would not yield an improvement if it produced too much noise. Further experiments with current MT systems are needed to assess whether this is actually the case. We are considering such tests for future research, using the Moses SMT system (Koehn et al., 2007).

Our representation had several possible advantages over pure parse-based methods. We used semantic features such as temporal, locative and transparent (whether a low-content words inherits its semantics) to help guide our alignment. The regularized structure, also, helped identify long-distance dependency relationships. We are also considering several improvements for our alignment-based rules: (1) using additional dictionary resources such as CATVAR (Habash and Dorr, 2003), so that cross-part-of speech alignments can be more readily recognized; (2) finding more optimal orderings for unaligned source language words. For example, the alignment-based method reordered *a bright star arising from China's policy* to *a bright arising from China 's policy star*, separating *bright* from *star*, even though *bright star* function as a unit; (3) incorporating and using multi-word bilingual dictionary entries.; (4) automatic methods for tuning parameters of our system that are currently hand-coded; (5) training MI on a much larger corpus; (6) investigating possible ways to merge the manual-rules with the alignment-based approach; and (7) performing similar experiments with English/Japanese bitexts.

We would expect both parse-based approaches and our system to handle mismatches that cover large distances better than more shallow approaches to reordering, e.g., (Crego and Mariño, 2006) in the same way that a full-parse handles constituent structure more completely than a chunker. In addition, we would expect our approach to work best in languages where there are large differences in word order, as these are exactly the cases that all predicate-argument structure is designed to handle well (they reduce apparent variation in structure). Towards this end we are currently working on a Japanese/English system. Obviously, the cost of developing GLARF (or similar) systems are high, require linguistic expertise and may not be possible for resource-poor languages. Nevertheless, we maintain that such systems are useful for many purposes and are therefore worth the cost. The GLARF system for English is available for download at `http://nlp.cs.nyu.edu/meyers/GLARF.html`.

Acknowledgments

This work was supported by NSF Grant IIS-0534700 Structure Alignment-based MT.

References

J. Bresnan and R. M. Kaplan. 1982. Syntactic Representation: Lexical-Functional Grammar: A Formal Theory for Grammatical Representation. In J. Bresnan, editor, *The Mental Representation of Grammatical Relations*. The MIT Press, Cambridge.

A. Bryl and J. van Genabith. 2010. f-align: An Open-Source Alignment Tool for LFG f-Structures. In *Proceedings of AMTA 2010*.

E. Charniak. 2001. Immediate-head parsing for language models. In *ACL 2001*, pages 116–123.

N. Chomsky. 1957. *Syntactic Structures*. Mouton, The Hague.

M. Collins, P. Koehn, and I. Kucerova. 2005. Clause Restructuring for Statistical Machine Translation. In *ACL 2005*.

B. A. Cowan. 2008. *A Tree-to-Tree Model for Statistical Machine Translation*. Ph.D. thesis, MIT.

J. M. Crego and J. B. Mariño. 2006. Integration of POS-tag-based source reordering into SMT decoding by an extended search graph. In *AMTA'06*.

C. Fellbaum, editor. 1998. *WordNet: An Electronic Lexical Database*. The MIT Press, Cambridge.

A. Fraser and D. Marcu. 2007. Measuring Word Alignment Quality for Statistical Machine Translation. *Computational Linguistics*, 33:293–303.

D. Gildea and D. Jurafsky. 2002. Automatic Labeling of Semantic Roles. *Computational Linguistics*, 28:245–288.

R. Grishman. 2000. Entity Annotation Guidelines. ftp://jaguar.ncsl.nist.gov/ace/phase1/edt_phase1_v2.2.pdf.

N. Habash and B. Dorr. 2003. CatVar: A Database of Categorial Variations for English. In *Proceedings of the MT Summit*, pages 471–474, New Orleans.

Z. Harris. 1968. *Mathematical Structures of Language*. Wiley-Interscience, New York.

J. R. Hobbs and R. Grishman. 1976. The Automatic Transformational Analysis of English Sentences: An Implementation. *International Journal of Computer Mathematics*, 5:267–283.

H. Ji and R. Grishman. 2006. Analysis and Repair of Name Tagger Errors. In *COLING/ACL 2006*, Sydney, Australia.

H. Ji, R. Grishman, D. Freitag, M. Blume, J. Wang, S. Khadivi, R. Zens, and H. Ney. 2009. Name Translation for Distillation. In *Global Autonomous Language Exploitation*. Springer.

P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. In *ACL 2007 Demonstration Session*, Prague.

P. Laplace. 1816. *Essai philosophique sur les probabilits*. Courcier Imprimeur, Paris.

Adam Meyers, Roman Yangarber, and Ralph Grishman. 1996. Alignment of Shared Forests for Bilingual Corpora. In *Proceedings of Coling 1996: The 16th International Conference on Computational Linguistics*, pages 460–465.

Adam Meyers, Roman Yangarber, Ralph Grishman, Catherine Macleod, and Antonio Moreno-Sandoval. 1998. Deriving Transfer Rules from Dominance-Preserving Alignments. In *Proceedings of Coling-ACL98: The 17th International Conference on Computational Linguistics and the 36th Meeting of the Association for Computational Linguistics*.

A. Meyers, R. Reeves, C. Macleod, R. Szekely, V. Zielinska, B. Young, and R. Grishman. 2004. Annotating Noun Argument Structure for NomBank. In *Proceedings of LREC-2004*, Lisbon, Portugal.

A. Meyers, M. Kosaka, N. Xue, H. Ji, A. Sun, S. Liao, and W. Xu. 2009. Automatic Recognition of Logical Relations for English, Chinese and Japanese in the GLARF Framework. In *SEW-2009 at NAACL-HLT-2009*.

E. Miltsakaki, A. Joshi, R. Prasad, and B. Webber. 2004. Annotating discourse connectives and their arguments. In A. Meyers, editor, *NAACL/HLT 2004 Workshop: Frontiers in Corpus Annotation*, pages 9–16, Boston, Massachusetts, USA, May 2 - May 7. Association for Computational Linguistics.

F. J. Och and H. Ney. 2000. Improved Statistical Alignment Models. In *ACL 2000*.

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

M. Palmer, D. Gildea, and P. Kingsbury. 2005. The Proposition Bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–106.

S. Petrov and D. Klein. 2007. Improved Inference for Unlexicalized Parsing. In *HLT-NAACL 2007*.

L. Shen, J. Xu, and R. Weischedel. 2008. A New String-to-Dependency Machine Translation Algorithm with a Target Dependency Language Model. In *ACL 2008*.

J. Wagner, D. Seddah, J. Foster, and J. van Genabith. 2007. C-Structures and F-Structures for the British National Corpus. In *Proceedings of the Twelfth International Lexical Functional Grammar Conference*, Stanford. CSLI Publications.

C. Wang, M. Collins, and P. Koehn. 2007. Chinese syntactic reordering for statistical machine translation. In *EMNLP-CoNLL 2007*, pages 737–745.

D. Wu and P. Fung. 2009. Semantic roles for smt: A hybrid two-pass model. In *HLT-NAACL-2009*, pages 13–16, Boulder, Colorado, June. Association for Computational Linguistics.

N. Xue and M. Palmer. 2003. Annotating the Propositions in the Penn Chinese Treebank. In *The Proceedings of the 2nd SIGHAN Workshop on Chinese Language Processing*, Sapporo.

N. Xue. 2008. Labeling Chinese Predicates with Semantic roles. *Computational Linguistics*, 34:225–255.

K. Yamada and K. Knight. 2001. A syntax-based statistical translation model. In *ACL*, pages 523–530.

Y. Zhang, R. Zens, and H. Ney. 2007. Chunk-Level Reordering of Source Language Sentences with Automatically Learned Rules for Statistical Machine Translation. In *Proc. of NAACL/HLT 2007*.

M. Zhang, H. Jiang, A. Aw, H. Li, C. L. Tan, and S. Li. 2008. A Tree Sequence Alignment-based Tree-to-Tree Translation Model. In *ACL 2008*.