# Word-level Alignment for Multilingual Resource Acquisition

**Adam Lopez**[*], **Michael Nossal**[*], **Rebecca Hwa**[*], **Philip Resnik**[*†]

[*]University of Maryland Institute for Advanced Computer Studies
[†]University of Maryland Department of Linguistics
College Park, MD 20742
{alopez, nossal, hwa, resnik}@umiacs.umd.edu

**Abstract**

We present a simple, one-pass word alignment algorithm for parallel text. Our algorithm utilizes synchronous parsing and takes advantage of existing syntactic annotations. In our experiments the performance of this model is comparable to more complicated iterative methods. We discuss the challenges and potential benefits of using this model to train syntactic parsers for new languages.

## 1 Introduction

*Word alignment* is an exercise commonly assigned to students learning a foreign language. Given a pair of sentences that are translations of each other, the students are asked to draw lines between words that mean the same thing.

In the context of multi-lingual natural language processing, word alignment (more simply, *alignment*) is also a necessary step for many applications. For instance, it is required in the parameter estimation step for training statistical translation models (Al-Onaizan et al., 1999; Brown et al., 1990; Melamed, 2000). Alignments are also useful for foreign language resource acquisition. Yarowsky and Ngai (2001) use an alignment to project part-of-speech (POS) tags from English to Chinese, and use the resulting noisy corpus to train a reliable Chinese POS tagger. Their result suggests that is worthwhile to consider more ambitious endeavors in resource acquisition.

Creating a syntactic treebank (e.g., the Penn Treebank Project (Marcus et al., 1993)) is time-consuming and expensive. As a consequence, state-of-the-art stochastic parsers which rely on such treebanks exist only in languages such as English for which they are available. If syntactic annotation could be projected from English to a language for which no treebank has been developed, then the treebank bottleneck may be overcome (Cabezas et al., 2001).

In principle, the success of treebank acquisition in this manner depends on a few key assumptions. The first assumption is that syntactic relationships in one language can be directly projected to another language using an accurate alignment. This theory is explored in Hwa et al. (2002b). A second assumption is that we have access to a reliable English parser and a word aligner. Although high-quality English parsers are available, high-quality aligners are more difficult to come by. Most alignment research has out of necessity concentrated on unsupervised methods. Even the best results are much worse than alignments created by humans. Therefore, this paper focuses on producing alignments that are tailored to the aims of syntactic projection. In particular, we propose a novel alignment model that, given an English sentence, its dependency parse tree, and its translation, simultaneously generates alignments and a dependency tree for the translation.

Our alignment model aims to improve alignment accuracy while maintaining sensitivity to constraints imposed by the syntactic transfer task. We hypothesize that the incorporation of syntactic knowledge into the alignment model will result in higher quality alignments. Moreover, by generating alignments and parse trees simultaneously, the alignment algorithm avoids irreconcilable errors in the projected trees such as crossing dependencies. Thus, our two objectives complement each other.

To verify these hypotheses, we have performed a suite of experiments, evaluating our algorithm on the quality of the resulting alignments and projected parse trees for English and Chinese sentence pairs. Our initial experiments demonstrate that our approach produces alignments and dependency trees whose quality is comparable to those produced by current state-of-the art systems.

We acknowledge that the strong assumptions we have stated for the success of treebank acquisition do not always hold true (Hwa et al., 2002a; Hwa et al., 2002b). Therefore, it will be necessary to devise a training algorithm that learns syntax even in the face of substantial noise introduced by failures in these assumptions. Although this last point is beyond the scope of this paper, we will allude to potential syntactic transfer approaches that are possible with our system, but infeasible under other approaches.

## 2 Background

Synchronous parsing appears to be the best model for syntactic projection. Synchronous parsing models the translation process as dual sentence generation in which a word and its translation in the other sentence are generated in lockstep. Translation pairs of both words and phrases are

This article appears in *Proceedings of the 2002 LREC Workshop on Linguistic Knowledge Acquisition and Representation – Bootstrapping Annotated Language Data.*, pp. 34–42, Jun. Las Palmas, Spain. 

generated in a manner consistent with the syntax of their respective languages, but in a way that expresses the same relationship to the rest of the sentence. Thus, alignment and syntax are produced simultaneously and induce mutual constraints on each other. This model is ideal for the pursuit of our objectives, because it captures our complementary goals in an elegant theoretical framework.

Synchronous parsing requires both parses to adhere to the constraints of a given monolingual parsing model. If we assume context-free grammars, then each parse must be context-free. If we assume dependency grammars, then each parse must observe the planarity and connectivity constraints typical of such grammars (e.g. Sleator and Temperley (1993)).

In contrast, many alignment models (Melamed, 2000; Brown et al., 1990) rely on a bag-of-words model. This model presupposes no structural constraints on either input sentence beyond its linear order. To see why this type of model is problematic for syntactic transfer, consider what happens when syntax subsequently interacts with its output. Projecting dependencies across such an alignment may result in a dependency tree that violates planarity and connectivity constraints (Figure 1).

Once the fundamental assumptions of the syntactic model have been breached, there is no clear way to recover. For this reason, we would prefer not to use bag-of-words alignment models, although in many respects they remain state-of-the-art for alignment.

A canonical example of synchronous parsing is the Stochastic Inversion Transduction Grammar (SITV) (Wu, 1995). The SITV model imposes the constraints of context-free grammars on the synchronous parsing environment. However, we regard context-free grammars as problematic for our task, because recent statistical parsing models (Charniak, 2000; Collins, 1999; Ratnaparkhi, 1999) owe much of their success to ideas inherent to dependency parsing. We therefore adopt an algorithm described in Alshawi and Douglas (2000).[1] Their algorithm constructs synchronous dependency parses in the context of a domain-specific speech-to-speech translation system. In their system, synchronous parsing only enforces a contiguity constraint on phrasal translations. The actual syntax of the sentence is not assumed to be known. Nevertheless, their model is a synchronous parser for dependency syntax, and we adopt it for our purposes.

## 3 Our Modified Alignment Algorithm

We introduce parse trees as an optional input to the algorithm of Alshawi and Douglas (2000). We require that

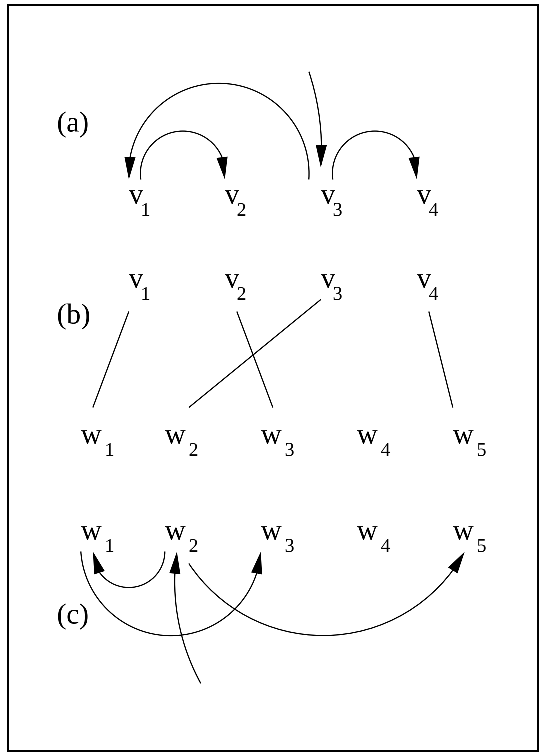


Figure 1: Violation of dependency grammar constraints caused by projecting a dependency parse across a bag-of-words alignment. Combining the syntax of (a) with the alignment of (b) produces the syntax of (c). In this example, the link $(w_1, w_3)$ crosses the link $(w_2, w_5)$ violating the planarity constraint. The word $w_4$ is unconnected, violating the connectivity constraint.

output dependency trees conform to dependency trees that are provided as input. If no parse tree is provided, our algorithm behaves identically to that of Alshawi and Douglas (2000).

### 3.1 Definitions

Our input is a parallel corpus that has been segmented into sentence pairs. We represent a sentence pair as the pair of word sequences ($V = v_1 ... v_m$, $W = w_1 ... w_n$). The algorithm iterates over the sentence pairs producing alignments.

We define a dependency parse as a rooted tree in which all words of the sentence appear once, and each node in the tree is such a word (Figure 2). An in-order traversal of the tree produces the sentence. A word is said to be *modified* by any words that appear as its children in the tree; conversely, the parent of a word is known as its *headword*. A word is said to *dominate* the span of all words that are descended from it in the tree, and is like-

[1] An alternative to dependency grammar is the richer formalism of Synchronized Tree-Adjoining Grammar (TAG) (Shieber and Schabes, 1990). However, Synchronized TAG raises issues of computational complexity and has not yet been exploited in a stochastic setting.

This article appears in *Proceedings of the 2002 LREC Workshop on Linguistic Knowledge Acquisition and Representation – Bootstrapping Annotated Language Data.*, pp. 34–42, Jun. Las Palmas, Spain. 

wise known as the *headword* of that span.[2] Subject to these constraints, the dependency parse of $V$ is expressed as a function $p_V : \{1...m\} \rightarrow \{0...m\}$ which defines the headword of each word in the dependency graph. The expression $p_V(i) = 0$ indicates that word $v_i$ is the root node of the graph (the headword of the sentence). The dependency parse of $W$, $p_W : \{1...n\} \rightarrow \{0...n\}$ is defined in the same way.

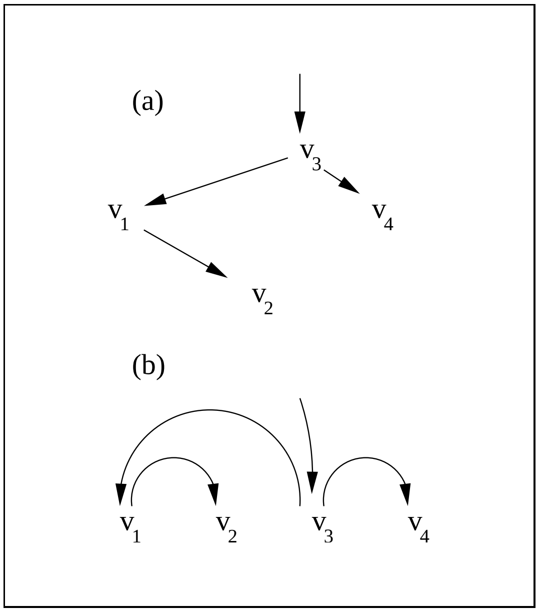


Figure 2: A dependency parse. In (a) the sentence is depicted in a tree form that makes the dominance and headword relationships clear ($v_3$ is the headword of the sentence). In (b) the same tree is depicted in more familiar sentence form, with the links drawn above the words.

An alignment is expressed as a function $a : \{1...m\} \rightarrow \{0...n\}$ in which $a(i) = j$ indicates that word $v_i$ of $V$ is aligned with word $w_j$ of W. The case in which $a(i) = 0$ denotes *null alignment* (i.e. the word $v_i$ does not correspond to any word in $W$). Under the constraints of synchronous parsing, we require that if $a(i) \neq 0$, then $p_W(a(i)) = a(p_V(i))$. In other words, the headword of a word's translation is the translation of the word's headword (Figure 3). We also require that the analogous condition hold for the inverse alignment map $a^{-1} : \{1...n\} \rightarrow \{0...m\}$.

### 3.2 Algorithm Details

Our algorithm (Appendix) is a bottom-up dynamic programming procedure. It is initialized by considering all possible alignments of one word to another word or to null. Alshawi and Douglas (2000) considered alignments of two words to one or no words, but we found in our evaluations

[2]Elsewhere, the terms *connectivity* and *planarity* are used to define these constraints.

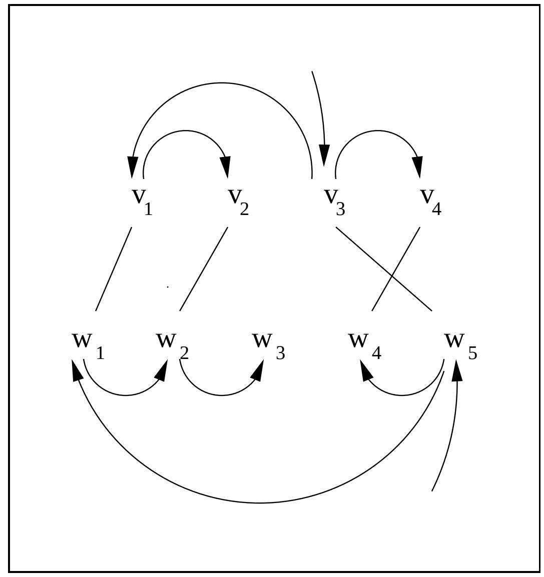

Figure 3: Synchronous dependency parses. Notice that all dependency links are symmetric across the alignment. In addition, the unaligned word $w_3$ is connected in the parse of $W$.

that restricting the initialization step to one word produced better results. In fact, Melamed (2000) argues in favor of exclusively one-to-one alignments. However, we may later explore in more detail the effects of initializing from multiword alignments.

As in Alshawi and Douglas (2000) each possible one-to-one alignment is scored using the $\phi^2$ metric (Gale and Church., 1991), which is used to compute the correlation between $v_i \in V$ and $w_j \in W$ over all sentence pairs $(V, W)$ in the corpus. Sentence co-occurrence counts are not the only possible data set with which we can use this metric. Therefore, we denote this type of initialization by $\phi^2_A$ to distinguish from a case we consider in Section 4.7, in which we use $\phi^2$ initialized from counts of Giza++ alignment links. The latter case is denoted by $\phi^2_G$.

To compute alignments of larger spans, the algorithm combines adjacent sub-alignments. During this step, one sub-alignment becomes a modifier phrase. Interpreting this in terms of dependency parsing, the aligned headwords of the modifier phrase become modifiers of the aligned headwords of the other phrase. At each step, the score of the alignment is computed. Following Alshawi and Douglas (2000) we simply add the score of the sub-alignments. Thus the overall score of any aligned subphrase can be computed as follows.

$$\sum_{(i,j):a(i)=j} \phi^2(v_i, w_j)$$

The output of the algorithm is simply the highest-scoring alignment that covers the entire span of both $V$ and

This article appears in *Proceedings of the 2002 LREC Workshop on Linguistic Knowledge Acquisition and Representation – Bootstrapping Annotated Language Data.*, pp. 34–42, Jun. Las Palmas, Spain. © European Languages Resources Association

$W$.

### 3.3 Treatment of Null Alignments

Null alignments present a few practical issues. For experiments involving $\phi_A^2$, we adopt the practice of counting a null token in the shorter sentence of each pair.[3] An alternative solution to this problem would involve initialization from a word association model that explicitly handles nulls, such as that of Melamed (2000).

An implication of the synchronous parsing constraint given in Section 3.1 is that null aligned words must be leaf words within their respective dependency graphs. In certain cases this may not lead to the best synchronized parse. We remove this condition. Effectively, we consider each sentence to consist of the same number of tokens, some of which may be null tokens. (usually, this will introduce null tokens into only the shorter sentence, but not necessarily). The null tokens behave like words with regards to the synchronous parsing constraint, but they do not impact phrase contiguity.[4] In only the resulting surface dependency graphs, we remove null tokens by contracting all edges between the null token and its parent and naming the resultant node with the word on the parent node. Recall from graph theory that contraction is an operation whereby an edge is removed and the nodes at its endpoints are conflated. [5] Thus, words that modify a null token are interpreted as modifiers of the the null token's headword. This is illustrated in Figure 4. One important implication of this is that we can only allow a null token to be the headword of the sentence if it has a single modifier. Otherwise, the result of the graph contraction would not be a rooted tree. We found that this treatment of null alignments resulted in a slight improvement in alignment results.

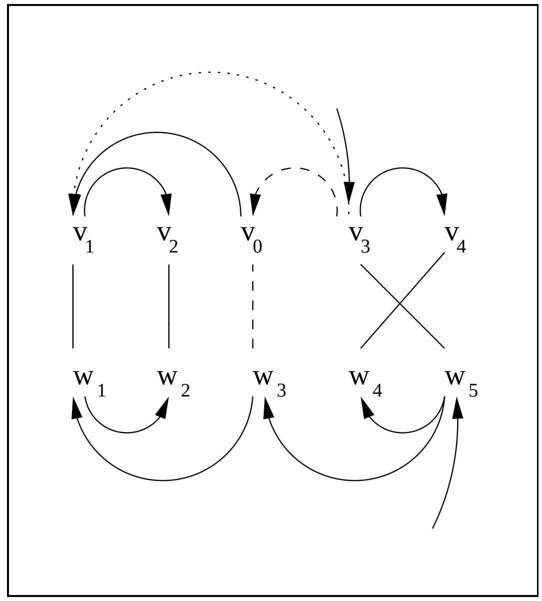


Figure 4: Effect of null words on synchronous parses. In this case, word $w_3$ has been aligned to the null token $v_0$. However, $v_0$ can still dominate other words in the parse of $V$. Once the structure has been completed, the edge between $v_0$ and $v_3$ (indicated by the dashed line) will contract. This will cause the dependency between $v_1$ and $v_0$ to become the inferred dependency (indicated by the dotted line) between $v_1$ and $v_3$.

### 3.4 Analysis

In the case that there are no parses available, the computational complexity of the algorithm is $O(m^3n^3)$, but with a parse of $V$ (and an efficient enumeration of the subphrase combinations allowed by the parse) the complexity reduces to $O(m^3n)$. If both parses are available the complexity would be reduced to $O(mn)$.

It is important to note that as it is presented, our algorithm does not search the entire space of possible alignment/tree combinations. Melamed observes that two modifications are required to accomplish this.[6] The first modification entails the addition of four new loop parameters to enumerate the possible headwords of the four monolingual subspans. These additional parameters add a factor of $O(m^2n^2)$. Second, Melamed points out that for a small subset of legal structures, it must be possible to combine subphrases that are not adjacent to one another. The most efficient solution to this problem adds two more parameters, for a total of $O(m^6n^6)$. The best known optimization reduces the total complexity to $O(m^5n^5)$. This is far too complex for a practical implementation, so we chose to use the original $O(m^3n^3)$ algorithm for our evaluations. Thus we recognize that our algorithm does not search the entire space of synchronous parses. It inherently incorporates a greedy heuristic, since for each subphrase, it considers only the most likely headword.

## 4 Evaluation

We have performed a suite of experiments to evaluate our alignment algorithm. The qualities of the resulting alignments and dependency parse trees are quantified by comparisons with correct human-annotated parses. We compare the alignment output of our algorithm with that of the basic algorithm described in Alshawi and Douglas (2000) and the well-known IBM statistical model described in Brown et al. (1990) using the freely available implementation (Giza++) described in Al-Onaizan et al. (1999). We also compare the output dependency trees against several

[3]Srinivas Bangalore, personal communication.

[4]a null token is considered to be contiguous with any other subphrase – another way to view this is that a null token is an unseen word that may appear at any location in the sentence in order to satisfy contiguity constraints.

[5]see e.g., Gross and Yellen (1999)

[6]I. Dan Melamed, personal communication.

| Synchronous Parsing Method | AP | AR | AF | CTP |
|---|---|---|---|---|
| sim-Alshawi ($\phi_A^2$) | 40.6 | 36.5 | 38.4 | 18.5 |
| sim-Alshawi ($\phi_A^2$) + English parse | 43.8 | 39.3 | 41.4 | 39.9 |
| sim-Alshawi ($\phi_A^2$) + English parse + Chinese bigrams | 42.9 | 38.5 | 40.6 | 39.4 |
| sim-Alshawi ($\phi_A^2$) + both bigrams | 41.5 | 37.3 | 39.3 | 16.5 |
| Giza++ initialization ($\phi_G^2$) | 51.2 | 45.9 | **48.4** | 11.6 |
| Giza++ initialization ($\phi_G^2$)+ English parse | 49.6 | 44.6 | 47.0 | **44.7** |

| Baseline Method | AP | AR | AF | CTP |
|---|---|---|---|---|
| Same Order Alignment | 15.7 | 14.1 | 14.8 | NA |
| Random Alignment (avg scores) | 7.8 | 7.0 | 7.4 | NA |
| Forward-chain | NA | NA | NA | 37.3 |
| Backward-chain | NA | NA | NA | 12.9 |
| Giza++ | 68.7 | 40.9 | **51.3** | NA |
| Hwa et al. (2002a) | NA | NA | NA | **44.1** |

Table 1: Alignment Results for All Methods.
AP = Alignment Precision. AR = Alignment Recall. AF = Alignment F-Score. CTP = Chinese Tree Precision.
All scores are reported as percentages of 100.
The best scores in each table appear in bold.

baselines and against projected dependency trees created in the manner described in (Hwa et al., 2002a). We found that our model, which combines cross-lingual statistics with syntactic annotation, produces alignments and trees that are are comparable to the best results of other methods.

### 4.1 Data Set

The language pair we have focused on for this study is English-Chinese. The training corpus consists of around 56,000 sentence pairs from the Hong Kong News parallel corpus. Because the training corpus is solely used for word co-occurrence statistics, no annotation is performed on it.

The development set was constructed by obtaining manual English translations for 47 Chinese sentences of 25 words or less, taken from sections 001-015 of the Chinese Treebank (Xia et al., 2000). A separate test set, consisting of 46 Chinese sentences of 25 words or less, was constructed in a similar fashion.[7] To obtain correct English parses, we used a context-free parser (Collins, 1999) and converted its output to dependency format. To obtain correct Chinese parses, Chinese Treebank trees were converted to dependency format. Both sets of parses were hand-corrected. The correct alignments for the development and test set were created by two native Chinese speakers using annotation software similar to that described in Melamed (1998).

[7]These sentences have already been manually translated into English as part of the NIST MT evaluation preview (See http://www.nist.gov/speech/tests/mt/). The sentences were taken from sections 038, 039, 067, 122, 191, 207, 249.

### 4.2 Metrics for evaluating alignments

As a measure of alignment accuracy, we report Alignment Precision ($AP$) and Alignment Recall ($AR$) figures. These are computed by by comparing the alignment links made by the system with the links in the correct alignment. We denote the set of guessed alignment links by $G_a$ and the set of correct alignment links by $C_a$. Precision is given by $AP = \frac{|C_a \cap G_a|}{|G_a|}$. Recall is given by $AR = \frac{|C_a \cap G_a|}{|C_a|}$. We also compute the F-score ($AF$), which is given by $AF = \frac{2 \cdot AP \cdot AR}{AP + AR}$. Null alignments are ignored in all computations. Our evaluation metric is similar to that of Och and Ney (2000).

### 4.3 Metrics for evaluating projected parse trees

As a measure of induced dependency tree accuracy, we report unlabeled Chinese Tree Precision ($CTP$). This is computed by comparing the output dependency tree with the correct dependency trees. We denote the set of guessed dependency links by $G_p$ and the set of correct alignment links by $C_p$. A small number of words (mostly punctuation) were not linked to any parent word in the correct parse; links containing these words are not included in either $C_p$ or $G_p$. Precision is given by $CTP = \frac{|C_p \cap G_p|}{|G_p|}$. For dependency trees, $|C_p| = |G_p|$, since each word contributes one link relating it to its headword. Thus, recall is the same as precision for our purposes.

### 4.4 Baseline Results

We first present the scores of some naïve algorithms as a baseline in order to provide a lower bound for our results. The results of the baseline experiments are included

This article appears in *Proceedings of the 2002 LREC Workshop on Linguistic Knowledge Acquisition and Representation – Bootstrapping Annotated Language Data.*, pp. 34–42, Jun. Las Palmas, Spain. © European Languages Resources Association

with all other results in Table 1. Our first baseline (Same Order Alignment) simply maps character $v_i$ in the English sentence to character $w_i$ in the Chinese sentence, or $w_n$ in the case of $i > n$. Our second baseline (Random Alignment), randomly aligns word $v_i$ to word $w_j$ subject to the constraint that no words are multiply aligned. We report the average scores over 100 runs of this baseline. The best Random Alignment F-score was 10.0% and the worst was 5.3% with a standard deviation of 0.9%.

For parse trees, we use two simple baselines. In the first (Forward-Chain), each word modifies the word immediately following it, and the last word is the headword of the sentence. For the second baseline (Backward-Chain), each word modifies the word immediately preceding it, and the first word is the headword of the sentence. No alignment was performed for these baselines.

The remaining baselines relate to the Giza++ algorithm. Giza++ produces the best word alignments. For reasons described previously, Giza++ alignments do not combine easily with syntax. However, Hwa et al. (2002a) contains an investigation in which trees output from a projection across Giza++ alignment are modified using several heuristics, and subsequently improved using linguistic knowledge of Chinese. We report the Chinese Tree Precision obtained by this method.

### 4.5 Synchronous Parsing Results

Our first set of alignments combines the $\phi_A^2$ cross-lingual co-occurrence metric described previously with either English parse or no parse trees. In this set, $\phi_A^2$ with no parse is nearly identical to the approach described in Alshawi and Douglas (2000) (excepting our treatment of null alignments). Thus, it serves as a useful point of comparison for runs that make use of other information. In Table 1 we refer to it as sim-Alshawi.

What we find is that incorporating parse trees results in a modest improvement over the baseline approach of sim-Alshawi. Why aren't the improvements more substantial? One observation is that using parses in this manner results in only passive interaction with the cross-lingual $\phi_A^2$ scores. In other words, the parse filters out certain alignments, but cannot in any other way counteract the biases inherent in the word statistics. Nevertheless, it appears to be modest progress.

### 4.6 Results of Using Bigrams to Approximate Parses

The results suggest that using parses to constrain the alignment is helpful. It is possible that using both parses would result in a more substantial improvement. However, we have already stated that we are interested in the case of asynchronous resources. Under this scenario, we only have access to one parse. Is there some way that we can approximate syntactic constraints of a sentence without having access to its parse?

The parsers of (Charniak, 2000; Collins, 1999; Ratnaparkhi, 1999) make substantial use of *bilexical dependencies*. Bilexical dependencies capture the idea that linked words in a dependency parse have a statistical affinity for each other: they often appear together in certain contexts. We suspect that bigram statistics could be used as a proxy for actual bilexical dependencies.

We constructed a simple test of this theory: for each English sentence $V = v_1 ... v_m$ in the development set with parse $p_V : \{1...m\} \to \{0...m\}$, we first construct the set of all bigrams $B = \{(v_i, v_j) : 1 \leq i < j \leq m\}$. We then partitioned $B$ into two sets: bigrams of linked words, i.e. $L = \{(v_i, v_j) : (v_i, v_j) \in B; p_V(v_i) = v_j$ or $p_V(v_j) = v_i\}$ and unlinked words $U = B - L$. We used the Bigram Statistics Package (Pedersen, 2001), to collect bigram statistics over the entire dev/train corpus and compute the average statistical correlation of each set using a variety of metrics (loglikelihood, dice, $\chi^2$, $\phi^2$). The results indicated that bigrams in the linked set $L$ were more correlated than those in the unlinked set $U$ under all metrics. We repeated this experiment with the development sentences in Chinese, with similar results. Although this is by no means a conclusive experiment, we took the results as an indication that using bigram statistics as an approximation of a parse might be helpful where no parse was actually available.

To incorporate bigram statistics into our alignment model, we modified the scoring function in the following manner: each time a dependency link is introduced between words and we do not have access to the source parse, we add into the alignment score the bigram score of the two words. The bigram score is based on the $\phi^2$ metric computed for bigram correlation. We call this $\phi_B^2$. The resulting alignment score can now be given by the following formula.

$$\sum_{(i,j):a(i)=j} \phi_A^2(v_i, w_j) + \sum_{(i,j):i<j, p_W(i)=j \wedge p_W(j)=i} \phi_B^2(w_i, w_j)$$

Our results indicate that using Chinese bigram statistics in conjunction with English parse trees in this manner results in a small decrease in the score along all measures. Nonetheless, there is an intuitively appealing interpretation of using bigrams in this way. The first is that the modification of the scoring function provides competitive interaction between parse information and cross-lingual statistics. The second is that if bigram statistics represent a weak approximation of syntax, then perhaps the iterative refinement of this statistic (e.g. by taking counts only over words that were linked in a previous iteration) would satisfy our objective of syntactic transfer.

### 4.7 Results of Using Better Word Statistics

Our results show that using parse information and coarse cross-lingual word statistics provides a modest boost over an approach using only the cross-lingual word statistics. We also decided to investigate what happens when we

This article appears in *Proceedings of the 2002 LREC Workshop on Linguistic Knowledge Acquisition and Representation – Bootstrapping Annotated Language Data.*, pp. 34–42, Jun. Las Palmas, Spain. 

seed our algorithm with better cross-lingual statistics

To test this, we initialize our co-occurrence counts from alignment links output by the Giza++ alignment of our corpus. We still use $\phi^2$ to compute the correlation. We call this $\phi^2_G$. Predictably, using the better word correlation statistics improves the quality of the alignment output in all cases. In this scenario, adding parse information does not seem to improve the alignment score. However, parse trees induced in this manner achieve a higher precision than any of the other methods. It outscores the baseline algorithms by a significant amount, and produces results comparable to the baseline of Hwa et al. (2002a). It is important to note, however, that the baseline of Hwa et al. (2002a) is achieved only after the application of numerous linguistic rules to the output of the Giza++ alignment. Additionally, the trees themselves may contain errors of the type described in Section 2. In contrast, our tree precision results directly from the application of our synchronous parsing algorithm, and all of the output trees are valid dependency parses.

## 5 Future Work

We believe that a fundamental advantage of our baseline model is its simplicity. Improving upon it will be considerably easier than improving upon a complex model such as the one described in Brown et al. (1990). Improvements may proceed along several possible paths. One path would involve reformulating the scoring functions in terms of statistical models (e.g. generative models). A natural complement to this path would be the introduction of iteration with the goal of improving the alignments and the accompanying models. In this approach, we could attempt to learn a coarse statistical model of the syntax of the low-density language after each iteration of the alignment. This information could in turn be used as evidence in the next iteration of the alignment model, hopefully improving its performance. Our results have already established a set of statistics that could be used in the initial iteration of such a task. The iterative approach resonates with an idea proposed in Yarowsky and Ngai (2001), regarding the use of learned part-of-speech taggers in subsequent alignment iterations.

An orthogonal approach would be the application of additional linguistic information. Our results indicated that syntactic knowledge can help improve alignment. Additional linguistic knowledge obtained from named-entity analyses, phrasal boundary detection, and part-of-speech tags might also improve alignment.

Although our output dependency trees represent definite progress, trees with such low precision cannot be used directly to train statistical parsers that assume correct training data (Charniak, 2000; Collins, 1999; Ratnaparkhi, 1999). There are two possible methods of improving upon the precision of this training data. The first is the use of noise-resistant training algorithms such as those described in (Yarowsky and Ngai, 2001). The second is the possibility of improving the precision yield by removing obviously bad training examples from the set. Unlike the baseline model, our word alignment model provides an obvious means of doing this. One possibility is to use a score gleaned from the alignment algorithm as a means of ranking dependency links, and removing links whose score is above some threshold. We hope that a dual approach of improving the precision of the training examples, while simultaneously reducing the sensitivity of the training algorithm, will result in the ability to train a reasonably accurate statistical parser for the new language. Our eventual objective is to train a parser in this manner.

## 6 Related work

Al-Onaizan et al. (1999), Brown et al. (1990) and Melamed (2000) focus on the description of statistical translation models based on the bag-of-words model. Alignment plays a crucial part in the parameter estimation methods of these models, but they remain problematic for syntactic transfer for reasons described in Section 2. The work of Hwa et al. (2002b) is an investigation into the combination of syntax with the output of this type of model. Och et al. (1999) presents a statistical translation model that performs phrasal translation, but it relies on shallow phrases that are discovered statistically, and makes no use of syntax. Yamada and Knight (2001) create a full-fledged syntax-based translation model. However, their model is unidirectional; it only describes the syntax of one sentence, and makes no provision for the syntax of the other. Wu (1995) presents a complete theory of synchronous parsing using a variant of context-free grammars, and exhibits several positive results, though not for syntax transfer. Alshawi and Douglas (2000) present the synchronous parsing algorithm on which our work is based. Much like the work on translation models, however, this work is interested in alignment primarily as a mechanism for training a machine translation system. Variations on the synchronous parsing algorithm appear in Alshawi et al. (2000a) and Alshawi et al. (2000b), but the algorithm of Alshawi and Douglas (2000) appears to be the most complete.

## 7 Conclusion

We have described a new approach to alignment that incorporates dependency parses into a synchronous parsing model. Our results indicate that this approach results in alignments whose quality is comparable to those produced by complicated iterative techniques. In addition, our approach demonstrates substantial promise in the task of learning syntactic models for resource-poor languages.

## 8 Acknowledgements

This work has been supported, in part, by ONR MURI Contract FCPO.810548265, DARPA/ITO Cooperative Agreement N660010028910, NSA Contract RD-02-5700 and Mitre Contract 010418-7712. The authors would like to thank I. Dan Melamed and Srinivas Bangalore for helpful discussions; Franz Josef Och for help with Giza++; and Lingling Zhang, Edward Hung, and Gina Levow for creating the gold standard annotations for the development and test data.

## 9 References

Yaser Al-Onaizan, Jan Curin, Michael Jahr, Kevin Knight, John Lafferty, I. Dan Melamed, Franz Josef Och, David Purdy, Noah A. Smith, and David Yarowsky. 1999. Statistical machine translation: Final report. In *Summer Workshop on Language Engineering*. John Hopkins University Center for Language and Speech Processing.

Hiyan Alshawi and Shona Douglas. 2000. Learning dependency transduction models from unannotated examples. *Philosophical Transactions of the Royal Society*, 358:1357–1372.

Hiyan Alshawi, Srinivas Bangalore, and Shona Douglas. 2000a. Learning dependency translation models as collections of finite state head transducers. *Computational Linguistics*, 26:1357–1372.

Hiyan Alshawi, Srinivasa Bangalore, and Shona Douglas. 2000b. Head transducer models for speech translation and their automatic acquisition from bilingual data. *Machine Translation*, 15:105–124.

Peter F. Brown, John Cocke, Stephen Della Pietra, Vincent J. Della Pietra, Fredrick Jelinek, John D. Lafferty, Robert L. Mercer, and Paul S. Roossin. 1990. A statistical approach to machine translation. *Computational Linguistics*, 16(2):79–85.

Clara Cabezas, Bonnie Dorr, and Philip Resnik. 2001. Spanish language processing at university of maryland: Building infrastructure for multilingual applications. In *Proceedings of the Second International Workshop on Spanish Language Processing and Language Technologies (SLPLT-2)*.

Eugene Charniak. 2000. A maximum-entropy-inspired parser. In *Proceedings of the 1st Meeting of the North American Chapter of the Association for Computational Linguistics*.

Michael Collins. 1999. *Head-Driven Statistical Models for Natural Language Parsing*. Ph.D. thesis, University of Pennsylvania.

William A. Gale and Kenneth W. Church. 1991. Identifying word correspondences in parallel texts. In *Proceedings of the Fourth DARPA Speech and Natural Language Processing Workshop*, pages 152–157.

Jonathan Gross and Jay Yellen, 1999. *Graph Theory and Its Applications*, chapter 7.5: Transforming a Graph by Edge Contraction, pages 263–266. Series on Discrete Mathematics and Its Applications. CRC Press.

Rebecca Hwa, Philip Resnik, and Amy Weinberg. 2002a. Breaking the resource bottleneck for multilingual parsing. In *Proceedings of the Workshop on Linguistic Knowledge Acquisition and Representation: Bootstrapping Annotated Language Data*. To appear.

Rebecca Hwa, Philip Resnik, Amy Weinberg, and Okan Kolak. 2002b. Evaluating translational correspondence using annotation projection. In *Proceedings of the 40th Annual Meeting of the ACL*. To appear.

Mitchell Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: the Penn Treebank. *Computational Linguistics*, 19(2):313–330.

I. Dan Melamed. 1998. Annotation style guide for the blinker project. Technical Report IRCS 98-06, University of Pennsylvania.

I. Dan Melamed. 2000. Models of translational equivalence among words. *Computational Linguistics*, 26(2):221–249, Jun.

Franz Josef Och and Hermann Ney. 2000. Improved statistical alignment models. In *Proceedings of the 38th Annual Meeting of the ACL*, pages 440–447.

Franz Josef Och, Christoph Tillmann, and Hermann Ney. 1999. Improved alignment models for statistical machine translation. In *Proceedings of the Joint Conference of Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 20–28, Jun.

Ted Pedersen. 2001. A decision tree of bigrams is an accurate predictor of word sense. In *Proceedings of the 2nd Meeting of the North American Chapter of the Association for Computational Linguistics*, pages 79–86, Jun.

Adwait Ratnaparkhi. 1999. Learning to parse natural language with maximum entropy models. *Machine Learning*, 34(1-3):151–175.

Stuart Shieber and Yves Schabes. 1990. Synchronous tree-adjoining grammars. In *Proceedings of the 13th International Conference on Computational Linguistics*, volume 3, pages 1–6.

Daniel Sleator and Davy Temperley. 1993. Parsing english with a link grammar. In *Third International Workshop on Parsing Technologies*, Aug.

Dekai Wu. 1995. Stochastic inversion transduction grammars, with application to segmentation, bracketing, and alignment of parallel corpora. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, pages 1328–1335, Aug.

Fei Xia, Martha Palmer, Nianwen Xue, Mary Ellen Ocurowski, John Kovarik, Fu-Dong Chiou, Shizhe Huang, Tony Kroch, and Mitch Marcus. 2000. Developing guidelines and ensuring consistency for chinese text annotation. In *Proceedings of the Second Language Resources and Evaluation Conference*, June.

Kenji Yamada and Kevin Knight. 2001. A syntax-based

This article appears in *Proceedings of the 2002 LREC Workshop on Linguistic Knowledge Acquisition and Representation – Bootstrapping Annotated Language Data.*, pp. 34–42, Jun. Las Palmas, Spain. 

statistical translation model. In *Proceedings of the Conference of the Association for Computational Linguistics*.

David Yarowsky and Grace Ngai. 2001. Inducing multilingual pos taggers and np bracketers via robust projection across aligned corpora. In *Proceedings of the 2nd Meeting of the North American Chapter of the Association for Computational Linguistics*, Jun.

This article appears in *Proceedings of the 2002 LREC Workshop on Linguistic Knowledge Acquisition and Representation – Bootstrapping Annotated Language Data.*, pp. 34–42, Jun. Las Palmas, Spain. 

## A Algorithm Pseudocode

*The following code does not address what constitutes a legal combination of subspans for an alignment. Legal subspans depend on constraints imposed by an input parse, if available. Otherwise, as in Alshawi and Douglas (2000), all possible combinations of subspans are legal. Regardless of what constitutes a legal subspan, the enumeration of spans must be done in a reasonable way. Small spans must be enumerated before larger spans that are constructed from them.*

*The variables* $i_V$ *and* $j_V$ *denote the span* $v_{i_V+1}...v_{j_V}$*, and* $p_V$ *denotes a partition of the span such that* $i_V \leq p_V \leq j_V$*. The variables* $i_W$*,* $j_W$*, and* $p_W$ *are defined analogously on* $W$*.*

*Our data structure is a chart* $\alpha$*, which contains cells indexed by* $i_V$*,* $j_V$*,* $i_W$*, and* $j_W$*. Each cell contains subfields* $phrase$*,* $modifierPhrase$*, and* $score$*.*

*Finally, we assume the existence of functions assocScore and score. The assocScore function computes the score of directly aligning to short spans of the sentence pair. In this paper, we use variations on the* $\phi^2$ *metric (Gale and Church., 1991) for this. The score function computes the score of combining two sub-alignments, assuming that the second sub-alignment becomes a modifier of the first. In this paper, we use one score function that simply adds the score of sub-alignments, and one that adds bigram correlation to the score of the sub-alignments. In principle, arbitrary scoring functions can be used.*

*initialize the chart*

for all legal combinations of $i_V$, $j_V$,$i_W$, and $j_W$

  $\alpha(i_V, j_V, i_W, j_W) = assocScore(v_{i_V+1}...v_{j_V}, w_{i_W+1}...w_{j_W})$

*complete the chart*

for all legal combinations of $i_V$, $j_V$, $p_V$, $i_W$, $j_W$, and $p_W$

  *consider the case in which aligned subphrases are in the same order in both languages.*

  $phrase = \alpha(i_V, p_V, i_W, p_W)$

  $modifierPhrase = \alpha(p_V, j_V, p_W, j_W)$

  $score =$score($phrase, modifierPhrase$)

  if $score > \alpha(i_V, j_V, i_W, j_W).score$ then

    $\alpha(i_V, j_V, i_W, j_W) =$ new subAlignment($phrase, modifierPhrase, score$)

  *consider the case in which the dominance relationship between these two phrases is reversed.*

  swap($phrase, modifierPhrase$)

  $score =$score($phrase, modifierPhrase$)

  if $score > \alpha(i_V, j_V, i_W, j_W).score$ then

    $\alpha(i_V, j_V, i_W, j_W) =$ new subAlignment($phrase, modifierPhrase, score$)

  *consider the case in which aligned subphrases are in the reverse order in each language.*

  $phrase = \alpha(i_V, p_V, p_W, j_W)$

  $modifierPhrase = \alpha(p_V, j_V, i_W, p_W)$

  $cost =$cost($phrase, modifierPhrase$)

  $score =$score($phrase, modifierPhrase$)

  if $score > \alpha(i_V, j_V, i_W, j_W).score$ then

    $\alpha(i_V, j_V, i_W, j_W) =$ new subAlignment($phrase, modifierPhrase, score$)

  *consider the case in which the dominance relationship between these two phrases is reversed.*

  swap($phrase, modifierPhrase$)

  $score =$score($phrase, modifierPhrase$)

  if $score > \alpha(i_V, j_V, i_W, j_W).score$ then

    $\alpha(i_V, j_V, i_W, j_W) =$ new subAlignment($phrase, modifierPhrase, score$)

return $\alpha(0, m, 0, n)$

This article appears in *Proceedings of the 2002 LREC Workshop on Linguistic Knowledge Acquisition and Representation – Bootstrapping Annotated Language Data.*, pp. 34–42, Jun. Las Palmas, Spain. 