An Algorithm for Simultaneously Bracketing Parallel Texts by Aligning Words

Dekai Wu
HKUST
Department of Computer Science
University of Science & Technology
Clear Water Bay, Hong Kong
dekai@cs.ust.hk

Abstract

We describe a grammarless method for simultaneously bracketing both halves of a parallel text and giving word alignments, assuming only a translation lexicon for the language pair. We introduce *inversion-invariant transduction grammars* which serve as generative models for parallel bilingual sentences with weak order constraints. Focusing on transduction grammars for bracketing, we formulate a normal form, and a stochastic version amenable to a maximum-likelihood bracketing algorithm. Several extensions and experiments are discussed.

1 Introduction

Parallel corpora have been shown to provide an extremely rich source of constraints for statistical analysis (e.g., Brown *et al.* 1990; Gale & Church 1991; Gale *et al.* 1992; Church 1993; Brown *et al.* 1993; Dagan *et al.* 1993; Dagan & Church 1994; Fung & Church 1994; Wu & Xia 1994; Fung & McKeown 1994). Our thesis in this paper is that the lexical information actually gives sufficient information to extract not merely word alignments, but also bracketing constraints for both parallel texts. Aside from purely linguistic interest, bracket structure has been empirically shown to be highly effective at constraining subsequent training of, for example, stochastic context-free grammars (Pereira & Schabes 1992; Black *et al.* 1993). Previous algorithms for automatic bracketing operate on monolingual texts and hence require more grammatical constraints; for example, tactics employing mutual information have been applied to tagged text (Magerman & Marcus 1990).

Algorithms for word alignment attempt to find the matching words between parallel sentences.[1] Although word alignments are of little use by themselves, they provide potential anchor points for other applications, or for subsequent learning stages to acquire more interesting structures. Our technique views word alignment and bracket annotation for both parallel texts as an integrated problem. Although the examples and experiments herein are on Chinese and English, we believe the model is equally applicable to other language pairs, especially those within the same family (say Indo-European).

Our bracketing method is based on a new formalism called an *inversion-invariant transduction grammar*. By their nature inversion-invariant transduction grammars overgenerate, because they permit too much constituent-ordering freedom. Nonetheless, they turn out to be very useful for recognition when the true grammar is not fully known. Their purpose is not to flag ungrammatical inputs; instead they assume that the inputs are grammatical, the aim being to extract structure from the input data, in kindred spirit with robust parsing.

2 Inversion-Invariant Transduction Grammars

A transduction grammar is a bilingual model that generates two output streams, one for each language. The usual view of transducers as having one input stream and one output stream is more appropriate for restricted or deterministic finite-state machines. Although finite-state transducers have been well studied, they are insufficiently powerful for bilingual models. The models we consider here are non-deterministic models where the two languages' role is symmetric.

We begin by generalizing transduction to context-free form. In a context-free transduction grammar, terminal symbols come in pairs that are emitted to separate output streams. It follows that each rewrite rule emits not one but two streams, and that every non-terminal stands for a class of derivable substring *pairs*. For example, in the rewrite rule

$$A \rightarrow B\ x/y\ C\ z/\epsilon$$

the terminal symbols x and z are symbols of the language L_1 and are emitted on stream 1, while the terminal symbol y is a symbol of the language L_2 and is emitted on stream 2. This rule implies that x/y must be a valid entry in the translation lexicon. A matched terminal symbol pair such as x/y is called a *couple*. As a special case, the null symbol ϵ in either language means that no output

[1] *Word matching* is a more accurate term than *word alignment* since the matchings may cross, but we follow the literature.

S	$\mapsto$	NP VP
PP	$\mapsto$	Prep NP
NP	$\mapsto$	Pro | Det Class NN
NN	$\mapsto$	Mod N | NN PP
VP	$\mapsto$	VV | VV NN | VP PP
VV	$\mapsto$	V | Adv V
Pro	$\mapsto$	I/我 | you/你
Det	$\mapsto$	a/一
Class	$\mapsto$	ϵ/本
Prep	$\mapsto$	for/給
N	$\mapsto$	book/書
V	$\mapsto$	took/拿了

Figure 1: Example IITG.

token is generated. We call a symbol pair such as x/ϵ an L_1-*singleton*, and ϵ/y an L_2-*singleton*.

We can employ context-free transduction grammars in simple attempts at generative models for bilingual sentence pairs. For example, pretend for the moment that the simple transduction grammar shown in Figure 1 is a context-free transduction grammar, ignoring the $\mapsto$ symbols that are in place of the usual $\rightarrow$ symbols. This grammar generates the following example pair of English and Chinese sentences in translation:

(1) a. [I [[took [a book]$_{\text{NP}}$]$_{\text{VP}}$ [for you]$_{\text{PP}}$]$_{\text{VP}}$]$_{\text{S}}$
b. [我 [[拿了 [一本書]$_{\text{NP}}$]$_{\text{VP}}$ [給你]$_{\text{PP}}$]$_{\text{VP}}$]$_{\text{S}}$

Each instance of a non-terminal here actually derives two substrings, one in each of the sentences; these two substrings are translation counterparts. This suggests writing the parse trees together:

(2) [I/我 [[took/拿了 [a/一 ϵ/本 book/書]$_{\text{NP}}$]$_{\text{VP}}$ [for/給 you/你]$_{\text{PP}}$]$_{\text{VP}}$]$_{\text{S}}$

The problem with context-free transduction grammars is that, just as with finite-state transducers, both sentences in a translation pair must share exactly the same grammatical structure (except for optional words that can be handled with lexical singletons). For example, the following sentence pair with a perfectly valid, alternative Chinese translation cannot be generated:

(3) a. [I [[took [a book]$_{\text{NP}}$]$_{\text{VP}}$ [for you]$_{\text{PP}}$]$_{\text{VP}}$]$_{\text{S}}$
b. [我 [[給你]$_{\text{PP}}$ [拿了 [一本書]$_{\text{NP}}$]$_{\text{VP}}$]$_{\text{VP}}$]$_{\text{S}}$

We introduce the device of an inversion-invariant transduction grammar (IITG) to get around the inflexibility of context-free transduction grammars. Productions are interpreted as rewrite rules just as with context-free transduction grammars, with one additional proviso: *when generating output for stream 2, the constituents on a rule's right-hand side may be emitted either left-to-right* (as usual) *or right-to-left* (in inverted order). We use $\mapsto$ instead of $\rightarrow$ to indicate this. Note that inversion is permitted at any level of rule expansion.

With this simple proviso, the transduction grammar of Figure 1 straightforwardly generates sentence-pair (3). However, the IITG's weakened ordering constraints now also permit the following sentence pairs, where some constituents have been reversed:

(4) a. *[I [[for you]$_{\text{PP}}$ [[a book]$_{\text{NP}}$ took]$_{\text{VP}}$]$_{\text{VP}}$]$_{\text{S}}$
b. [我 [[給你]$_{\text{PP}}$ [拿了 [一本書]$_{\text{NP}}$]$_{\text{VP}}$]$_{\text{VP}}$]$_{\text{S}}$

(5) a. *[[[you for]$_{\text{PP}}$ [[a book]$_{\text{NP}}$ took]$_{\text{VP}}$]$_{\text{VP}}$ I]$_{\text{S}}$
b. *[我 [[給你]$_{\text{PP}}$ [[書本一]$_{\text{NP}}$ 拿了]$_{\text{VP}}$]$_{\text{VP}}$]$_{\text{S}}$

As a bilingual generative linguistic theory, therefore, IITGs are not well-motivated (at least for most natural language pairs), since the majority of constructs do not have freely reversable constituents.

We refer to the direction of a production's L_2 constituent ordering as an *orientation*. It is sometimes useful to explicitly designate one of the two possible orientations when writing productions. We do this by distinguishing two varieties of concatenation operators on string-pairs, depending on the orientation. The operator $[\,]$ performs the "usual" pairwise concatenation so that $[AB]$ yields the string-pair (C_1, C_2) where $C_1 = A_1B_1$ and $C_2 = A_2B_2$. But the operator $\langle\rangle$ concatenates constituents on output stream 1 while reversing them on stream 2, so that $C_1 = A_1B_1$ but $C_2 = B_2A_2$. For example, the NP $\mapsto$ Det Class NN rule in the transduction grammar above actually expands to two standard rewrite rules:

$$
\begin{array}{lcl}
\text{NP} & \rightarrow & [\text{Det Class NN}] \\
\text{NP} & \rightarrow & \langle\text{Det Class NN}\rangle
\end{array}
$$

Before turning to bracketing, we take note of three lemmas for IITGs (proofs omitted):

Lemma 1 *For any inversion-invariant transduction grammar G, there exists an equivalent inversion-invariant transduction grammar G' where $T(G) = T(G')$, such that:*

1. *If $\epsilon \in L_1(G)$ and $\epsilon \in L_2(G)$, then G' contains a single production of the form $S' \rightarrow \epsilon/\epsilon$, where S' is the start symbol of G' and does not appear on the right-hand side of any production of G';*
2. *otherwise G' contains no productions of the form $A \rightarrow \epsilon/\epsilon$.*

Lemma 2 *For any inversion-invariant transduction grammar G, there exists an equivalent inversion-invariant transduction grammar G' where $T(G) = T(G')$, $T(G) = T(G')$, such that the right-hand side of any production of G' contains either a single terminal-pair or a list of nonterminals.*

Lemma 3 *For any inversion-invariant transduction grammar G, there exists an equivalent inversion transduction grammar G' where $T(G) = T(G')$, such that G' does not contain any productions of the form $A \rightarrow B$.*

3 Bracketing Transduction Grammars

For the remainder of this paper, we focus our attention on pure bracketing. We confine ourselves to *bracketing* *transduction grammars* (BTGs), which are IITGs where constituent categories are not differentiated. Aside from the start symbol S, BTGs contain only one non-terminal symbol, A, which rewrites either recursively as a string of A's or as a single terminal-pair. In the former case, the productions has the form $A \mapsto A^f$ where we use A^f to abbreviate $A \ldots A$, where the *fanout* f denotes the number of A's. Each A corresponds to a level of bracketing and can be thought of as demarcating some unspecified kind of syntactic category. (This same "repetitive expansion" restriction used with standard context-free grammars and transduction grammars yields bracketing grammars without orientation invariance.)

A *full* bracketing transduction grammar of degree f contains A productions of every fanout between 2 and f, thus allowing constituents of any length up to f. In principle, a full BTG of high degree is preferable, having the greatest flexibility to accomodate arbitrarily long matching sequences. However, the following theorem simplifies our algorithms by allowing us to get away with degree-2 BTGs. Later we will see how postprocessing restores the fanout flexibility (Section 5.2).

Theorem 1 *For any full bracketing transduction grammar T, there exists an equivalent bracketing transduction grammar T' in normal form where every production takes one of the following forms:*

$$\begin{array}{lcl} S & \mapsto & \epsilon/\epsilon \\ S & \mapsto & A \\ A & \mapsto & A\ A \\ A & \mapsto & x/y \\ A & \mapsto & x/\epsilon \\ A & \mapsto & \epsilon/y \end{array}$$

Proof By Lemmas 1, 2, and 3, we may assume T contains only productions of the form $S \mapsto \epsilon/\epsilon$, $A \mapsto x/y$, $A \mapsto x/\epsilon$, $A \mapsto \epsilon/y$, and $A \mapsto AA \ldots A$. For proof by induction, we need only show that any full BTG T of degree $f > 2$ is equivalent to a full BTG T' of degree $f-1$. It suffices to show that the production $A \mapsto A^f$ can be removed without any loss to the generated language, i.e., that the remaining productions in T' can still derive any string-pair derivable by T (removing a production cannot increase the set of derivable string-pairs). Let (E, C) be any string-pair derivable from $A \mapsto A^f$, where E is output on stream 1 and C on stream 2. Define E^i as the substring of E derived from the ith A of the production, and similarly define C^i. There are two cases depending on the concatenation orientation, but (E, C) is derivable by T' in either case.

In the first case, if the derivation used was $A \rightarrow [A^f]$, then $E = E^1 \ldots E^f$ and $C = C^1 \ldots C^f$. Let $(E', C') = (E^1 \ldots E^{f-1}, C^1 \ldots C^{f-1})$. Then (E', C') is derivable from $A \rightarrow [A^{f-1}]$, and thus $(E, C) = (E'E^f, C'C^f)$ is derivable from $A \rightarrow [A\ A]$. In the second case, the derivation used was $A \rightarrow \langle A^f \rangle$, and we still have $E = E^1 \ldots E^f$ but now $C = C^f \ldots C^1$. Now let $(E', C'') = (E^1 \ldots E^{f-1}, C^{f-1} \ldots C^1)$. Then (E', C'') is derivable from $A \rightarrow \langle A^{f-1} \rangle$, and thus $(E, C) = (E'E^f, C^f C'')$ is derivable from $A \rightarrow \langle A\ A \rangle$. □

A	$\mapsto$	accountable/負責
A	$\mapsto$	authority/管理局
A	$\mapsto$	financial/財政
A	$\mapsto$	secretary/司
A	$\mapsto$	to/向
A	$\mapsto$	will/將會
A	$\mapsto$	./。
A	$\mapsto$	be/ϵ
A	$\mapsto$	the/ϵ

Figure 2: Some relevant lexical productions.

4 Stochastic Bracketing Transduction Grammars

In a stochastic BTG (SBTG), each rewrite rule has a probability. Let a_f denote the probability of the A-production with fanout degree f. For the remaining (lexical) productions, we use $b(x, y)$ to denote $P[A \mapsto x/y|A]$. The probabilities obey the constraint that

$$\sum_f a_f + \sum_{x,y} b(x, y) = 1$$

For our experiments we employed a normal form transduction grammar, so $a_f = 0$ for all $f \neq 2$. The A-productions used were:

$$\begin{array}{llll} A & \stackrel{a_2}{\mapsto} & A\ A & \\ A & \stackrel{b(x,y)}{\mapsto} & x/y & \text{for all } x, y \text{ lexical translations} \\ A & \stackrel{b(x,\epsilon)}{\mapsto} & x/\epsilon & \text{for all } x \text{ English vocabulary} \\ A & \stackrel{b(\epsilon,y)}{\mapsto} & \epsilon/y & \text{for all } y \text{ Chinese vocabulary} \end{array}$$

The $b(x, y)$ distribution actually encodes the English-Chinese translation lexicon. As discussed below, the lexicon we employed was automatically learned from a parallel corpus, giving us the $b(x, y)$ probabilities directly. The latter two singleton forms permit any word in either sentence to be unmatched. A small ϵ-constant is chosen for the probabilities $b(x, \epsilon)$ and $b(\epsilon, y)$, so that the optimal bracketing resorts to these productions only when it is otherwise impossible to match words.

With BTGs, to parse means to build matched bracketings for sentence-*pairs* rather than sentences. This means that the adjacency constraints given by the nested levels must be obeyed in the bracketings of both languages. The result of the parse gives bracketings for both input sentences, as well as a bracket alignment indicating the corresponding brackets between the sentences. The bracket alignment includes a word alignment as a byproduct.

Consider the following sentence pair from our corpus:

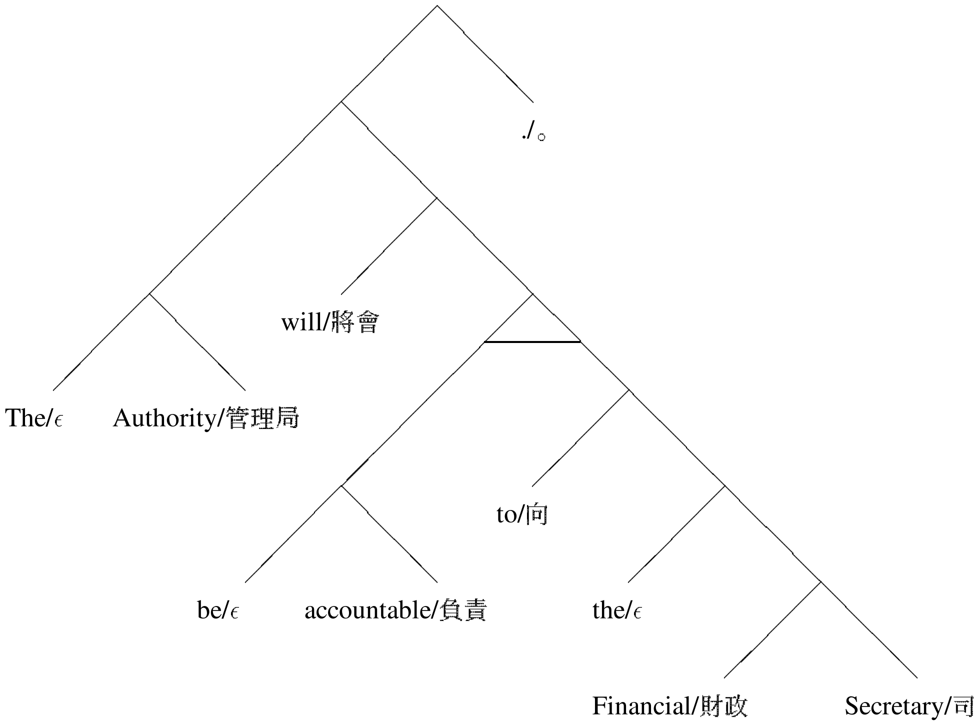

Figure 3: Bracketing tree.

(6) a. The Authority will be accountable to the Financial Secretary.

b. 管理局將會向財政司負責。

Assume we have the productions in Figure 2, which is a fragment excerpted from our actual BTG. Ignoring capitalization, an example of a valid parse that is consistent with our linguistic ideas is:

(7) [[[The/ϵ Authority/管理局] [will/將會 $\langle$[be/ϵ accountable/負責] [to/向 [the/ϵ [[Financial/財政 Secretary/司]]]]$\rangle$]] ./。]

Figure 3 shows a graphic representation of the same bracketing, where the $\langle\rangle$ level of bracketing is marked by the horizontal line. The English is read in the usual depth-first left-to-right order, but for the Chinese, a horizontal line means the right subtree is traversed before the left.

The $\langle\rangle$ notation concisely displays the common structure of the two sentences. However, the bracketing is clearer if we view the sentences monolingually, which allows us to invert the Chinese constituents within the $\langle\rangle$ so that only [ ] brackets need to appear:

(8) a. [[[The Authority] [will [[be accountable] [to [the [[Financial Secretary]]]]]]] .]

b. [[[[管理局] [將會 [[向 [[財政 司]]]] [負責]]]] 。]

In the monolingual view, extra brackets appear in one language whenever there is a singleton in the other language. If the goal is just to obtain brackets for monolingual sentences, the extra brackets can be discarded after parsing:

(9) [[[管理局] [將會 [向 [財政 司]] [負責]]] 。]

The basis of the bracketing strategy can be seen as choosing the bracketing that maximizes the (probabilistically weighted) number of words matched, subject to the BTG representational constraint, which has the effect of limiting the possible crossing patterns in the word alignment. A simpler, related idea of penalizing distortion from some ideal matching pattern can be found in the statistical translation (Brown *et al.* 1990; Brown *et al.* 1993) and word alignment (Dagan *et al.* 1993; Dagan & Church 1994) models. Unlike these models, however, the BTG aims to model constituent structure when determining distortion penalties. In particular, crossings that are consistent with the constituent tree structure are not penalized. The implicit assumption is that core arguments of frames remain similar across languages, and that core arguments of the same frame will surface adjacently. The accuracy of the method on a particular language pair will therefore depend upon the extent to which this language universals hypothesis holds. However, the approach is robust because if the assumption is violated, damage will be limited to dropping the fewest possible crossed word matchings.

We now describe how a dynamic-programming parser can compute an optimal bracketing given a sentence-pair and a stochastic BTG. In bilingual parsing, just as with ordinary monolingual parsing, probabilizing the grammar permits ambiguities to be resolved by choosing the maximum likelihood parse. Our algorithm is similar in spirit to the recognition algorithm for HMMs (Viterbi 1967).

Denote the input English sentence by $\mathbf{e}_1, \ldots, \mathbf{e}_T$ and the corresponding input Chinese sentence by $\mathbf{c}_1, \ldots, \mathbf{c}_V$. As an abbreviation we write $\mathbf{e}_{s..t}$ for the sequence of words $\mathbf{e}_{s+1}, \mathbf{e}_{s+2}, \ldots, \mathbf{e}_t$, and similarly for $\mathbf{c}_{u..v}$. Let $\delta_{stuv} = \max P[\mathbf{e}_{s..t}/\mathbf{c}_{u..v}]$ be the maximum probability of any derivation from A that successfully parses both substrings $\mathbf{e}_{s..t}$ and $\mathbf{c}_{u..v}$. The best parse of the sentence pair is that with probability $\delta_{0,T,0,V}$.

The algorithm computes $\delta_{0,T,0,V}$ following the recurrences below.[2] The time complexity of this algorithm is $\Theta(T^3V^3)$ where T and V are the lengths of the two sentences.

1. Initialization

$$\delta_{t-1,t,v-1,v} = b(\mathbf{e}_t/\mathbf{c}_v), \qquad \begin{array}{c} 1 \le t \le T \\ 1 \le v \le V \end{array}$$

2. Recursion

$$\begin{aligned} \delta_{stuv} &= \max[\delta^{[]}_{stuv}, \delta^{\langle\rangle}_{stuv}] \\ \theta_{stuv} &= \begin{cases} [\,] & \text{if } \delta^{[]}_{stuv} > \delta^{\langle\rangle}_{stuv} \\ \langle\rangle & \text{otherwise} \end{cases} \end{aligned}$$

where

$$\begin{aligned} \delta^{[]}_{stuv} &= \max_{\substack{s \le S \le t \\ u \le U \le v}} a_2\, \delta_{sSuU}\, \delta_{StUv} \\ \sigma^{[]}_{stuv} &= \arg_S \max_{\substack{s \le S \le t \\ u \le U \le v}} \delta_{sSuU}\, \delta_{StUv} \\ \upsilon^{[]}_{stuv} &= \arg_U \max_{\substack{s \le S \le t \\ u \le U \le v}} \delta_{sSuU}\, \delta_{StUv} \\ \delta^{\langle\rangle}_{stuv} &= \max_{\substack{s \le S \le t \\ u \le U \le v}} a_2\, \delta_{sSUv}\, \delta_{StuU} \\ \sigma^{\langle\rangle}_{stuv} &= \arg_S \max_{\substack{s \le S \le t \\ u \le U \le v}} \delta_{sSUv}\, \delta_{StuU} \\ \upsilon^{\langle\rangle}_{stuv} &= \arg_U \max_{\substack{s \le S \le t \\ u \le U \le v}} \delta_{sSUv}\, \delta_{StuU} \end{aligned}$$

3. Reconstruction Using 4-tuples to name each node of the parse tree, initially set $q_1 = (0, T, 0, V)$ to be the root. The remaining descendants in the optimal parse tree are then given recursively for any $q = (s, t, u, v)$ by:

$$\left.\begin{array}{l} \text{LEFT}(q) = (s, \sigma^{[]}_{stuv}, u, \upsilon^{[]}_{stuv}) \\ \text{RIGHT}(q) = (\sigma^{[]}_{stuv}, t, \upsilon^{[]}_{stuv}, v) \end{array}\right\} \text{ if } \theta_{stuv} = [\,]$$

$$\left.\begin{array}{l} \text{LEFT}(q) = (s, \sigma^{\langle\rangle}_{stuv}, \upsilon^{\langle\rangle}_{stuv}, v) \\ \text{RIGHT}(q) = (\sigma^{\langle\rangle}_{stuv}, t, u, \upsilon^{\langle\rangle}_{stuv}) \end{array}\right\} \text{ if } \theta_{stuv} = \langle\rangle$$

Several additional extensions on this algorithm were found to be useful, and are briefly described below. Details are given in Wu (1995).

[2] We are generalizing argmax as to allow arg to specify the index of interest.

4.1 Simultaneous segmentation

We often find the same concept realized using different numbers of words in the two languages, creating potential difficulties for word alignment; what is a single word in English may be realized as a compound in Chinese. Since Chinese text is not orthographically separated into words, the standard methodology is to first preprocess input texts through a segmentation module (Chiang *et al.* 1992; Lin *et al.* 1992; Chang & Chen 1993; Lin *et al.* 1993; Wu & Tseng 1993; Sproat *et al.* 1994). However, this seriously degrades our algorithm's performance, since the the segmenter may encounter ambiguities that are unresolvable monolingually and thereby introduce errors. Even if the Chinese segmentation is acceptable monolingually, it may not agree with the division of words present in the English sentence. Moreover, conventional compounds are frequently and unpredictably missing from translation lexicons, and this can further degrade performance.

To avoid such problems we have extended the algorithm to optimize the segmentation of the Chinese sentence in parallel with the bracketing process. Note that this treatment of segmentation does not attempt to address the open linguistic question of what constitutes a Chinese "word". Our definition of a correct "segmentation" is purely task-driven: longer segments are desirable if and only if no compositional translation is possible.

4.2 Pre/post-positional biases

Many of the bracketing errors are caused by singletons. With singletons, there is no cross-lingual discrimination to increase the certainty between alternative bracketings. A heuristic to deal with this is to specify for each of the two languages whether prepositions or postpositions are more common, where "preposition" here is meant not in the usual part-of-speech sense, but rather in a broad sense of the tendency of function words to attach left or right. This simple strategem is effective because the majority of unmatched singletons are function words that lack counterparts in the other language. This observation holds assuming that the translation lexicon's coverage is reasonably good. For both English and Chinese, we specify a prepositional bias, which means that singletons are attached to the right whenever possible.

4.3 Punctuation constraints

Certain punctuation characters give strong constituency indications with high reliability. "Perfect separators", which include colons and Chinese full stops, and "perfect delimiters", which include parentheses and quotation marks, can be used as bracketing constraints. We have extended the algorithm to precluded hypotheses that are inconsistent with such constraints, by initializing those entries in the DP table corresponding to illegal sub-hypotheses with zero probabilities. These entries are blocked from recomputation during the DP phase. As their probabilities always remain zero, the illegal bracketings can never participate in any optimal bracketing.

5 Postprocessing

5.1 A Singleton-Rebalancing Algorithm

We now introduce an algorithm for further improving the bracketing accuracy in cases of singletons. Consider the following bracketing produced by the algorithm of the previous section:

(10) [[The/ϵ [[Authority/管理局 [will/將會 $\langle$[be/ϵ accountable/負責] [to the/ϵ [ϵ/向 [Financial/財政 Secretary/司]]]$\rangle$]]]] ./。]

The prepositional bias has already correctly restricted the singleton "The/ϵ" to attach to the right, but of course "The" does not belong outside the rest of the sentence, but rather with "Authority". The problem is that singletons have no discriminative power between alternative bracket matchings—they only contribute to the ambiguity. However, we can minimize the impact by moving singletons as deep as possible, closer to the individual word they precede or succeed, by widening the scope of the brackets immediately following the singleton. In general this improves precision since wide-scope brackets are less constraining.

The algorithm employs a rebalancing strategy reminiscent of balanced-tree structures using left and right rotations. A left rotation changes a $(A(BC))$ structure to a $((AB)C)$ structure, and vice versa for a right rotation. The task is complicated by the presence of both $[\,]$ and $\langle\rangle$ brackets with both L_1- and L_2-singletons, since each combination presents different interactions. To be legal, a rotation must preserve symbol order on both output streams. However, the following lemma shows that any subtree can always be rebalanced at its root if either of its children is a singleton of either language.

Lemma 4 *Let x be a L_1 singleton, y be a L_2 singleton, and A, B, C be arbitrary constituent subtrees. Then the following properties hold for the $[\,]$ and $\langle\rangle$ operators:*

(Associativity)

$$[A[BC]] \rightleftharpoons [[AB]C]$$

$$\langle A\langle BC\rangle\rangle \rightleftharpoons \langle\langle AB\rangle C\rangle$$

(L_1-singleton bidirectionality)

$$[Ax] \rightleftharpoons \langle Ax\rangle$$

$$[xA] \rightleftharpoons \langle xA\rangle$$

(L_2-singleton flipping commutativity)

$$[Ay] \rightleftharpoons \langle yA\rangle$$

$$[yA] \rightleftharpoons \langle Ay\rangle$$

(L_1-singleton rotation properties)

$$[x\langle AB\rangle] \rightleftharpoons \langle x\langle AB\rangle\rangle \rightleftharpoons \langle\langle xA\rangle B\rangle \rightleftharpoons \langle[xA]B\rangle$$

$$\langle x[AB]\rangle \rightleftharpoons [x[AB]] \rightleftharpoons [[xA]B] \rightleftharpoons [\langle xA\rangle B]$$

$$[\langle AB\rangle x] \rightleftharpoons \langle\langle AB\rangle x\rangle \rightleftharpoons \langle A\langle Bx\rangle\rangle \rightleftharpoons \langle A[Bx]\rangle$$

$$\langle[AB]x\rangle \rightleftharpoons [[AB]x] \rightleftharpoons [A[Bx]] \rightleftharpoons [A\langle Bx\rangle]$$

(L_2-singleton rotation properties)

$$[y\langle AB\rangle] \rightleftharpoons \langle\langle AB\rangle y\rangle \rightleftharpoons \langle A\langle By\rangle\rangle \rightleftharpoons \langle A[yB]\rangle$$

$$\langle y[AB]\rangle \rightleftharpoons [[AB]y] \rightleftharpoons [A[By]] \rightleftharpoons [A\langle yB\rangle]$$

$$[\langle AB\rangle y] \rightleftharpoons \langle y\langle AB\rangle\rangle \rightleftharpoons \langle\langle yA\rangle B\rangle \rightleftharpoons \langle[Ay]B\rangle$$

$$\langle[AB]y\rangle \rightleftharpoons [y[AB]] \rightleftharpoons [[yA]B] \rightleftharpoons [\langle Ay\rangle B]$$

The method of Figure 4 modifies the input tree to attach singletons as closely as possible to couples, but remaining consistent with the input tree in the following sense: singletons cannot "escape" their immediately surrounding brackets. The key is that for any given subtree, if the outermost bracket involves a singleton that should be rotated into a subtree, then exactly one of the singleton rotation properties will apply. The method proceeds depth-first, sinking each singleton as deeply as possible. For example, after rebalancing, sentence (10) is bracketed as follows:

(11) [[[[The/ϵ Authority/管理局] [will/將會 $\langle$[be/ϵ accountable/負責] [to the/ϵ [ϵ/向 [Financial/財政 Secretary/司]]]$\rangle$]]] ./。]

5.2 Flattening the Bracketing

Because the BTG is in normal form, each bracket can only hold two constituents. This improves parsing efficiency, but requires overcommitment since the algorithm is always forced to choose between $(A(BC))$ and $((AB)C)$ structures even when no choice is clearly better. In the worst case, both sentences might have perfectly aligned words, lending no discriminative leverage whatsoever to the bracketer. This leaves a very large number of choices: if both sentences are of length $l = m$, then there are $\binom{2l}{l}\frac{1}{l+1}$ possible bracketings with fanout 2, none of which is better justified than any other. Thus to improve accuracy, we should reduce the specificity of the bracketing's commitment in such cases.

We implement this with another postprocessing stage. The algorithm proceeds bottom-up, eliminating as many brackets as possible, by making use of the associativity equivalences $[ABC] \rightleftharpoons [A[BC]] \rightleftharpoons [[AB]C]$ and

```
SINK-SINGLETON(node)
  if node is not a leaf
    if a rotation property applies at node
      apply the rotation to node
      child ← the child into which the singleton
                was rotated
      SINK-SINGLETON(child)

REBALANCE-TREE(node)
  if node is not a leaf
    REBALANCE-TREE(left-child[node])
    REBALANCE-TREE(right-child[node])
    SINK-SINGLETON(node)
```

Figure 4: The singleton rebalancing schema.

[These/這些 arrangements/安排 will/ϵ ϵ/可 enhance/加強 our/我們 ⟨[ϵ/的 ability/能力] [to/ϵ ϵ/日後 maintain/維持 monetary/金融 stability/穩定 in the years to come/ϵ]⟩ ./。]

[The/ϵ Authority/管理局 will/將會 ⟨[be/ϵ accountable/負責] [to the/ϵ ϵ/向 Financial/財政 Secretary/司]⟩ ./。]

[They/他們 ⟨ are/ϵ right/正確 ϵ/十分 to/ϵ do/做 ϵ/這樣 so/ϵ ⟩ ./。]

[⟨[Even/ϵ more/更 important/重要] [,/ϵ however/但]⟩ [,/ϵ ϵ/的, is/是 to make the very best of our/ϵ ϵ/善用香港 own/本身 ϵ/的 talent/人才] ./。]

[I/我 hope/ϵ ϵ/<>望 employers/僱主 will/會 make full/ϵ ϵ/充分善 use/用 [of/ϵ those/那些] ⟨⟨[ϵ/的工 who/人] [have acquired/ϵ ϵ/學到 new/新 skills/技能]⟩ [through/透過 this/這個 programme/計劃]⟩ ./。]

[I/我 have/已 <> at/ϵ length/詳細 ⟨ on/ϵ how/怎樣 we/我們 ϵ/講述⟩ [can/可以 boost/ϵ ϵ/促進 our/本港 ϵ/的 prosperity/繁榮] ./。]

Figure 5: Bracketing/alignment output examples. (<> = unrecognized input token.)

$\langle ABC\rangle \rightleftharpoons \langle A\langle BC\rangle\rangle \rightleftharpoons \langle\langle AB\rangle C\rangle$. The singleton bidirectionality and flipping commutativity equivalences (see Lemma 4) are also applied, whenever they render the associativity equivalences applicable.

The final result after flattening sentence (11) is as follows:

(12) [The/ϵ Authority/管理局 will/將會 ⟨[be/ϵ accountable/負責] [to the/ϵ ϵ/向 Financial/財政 Secretary/司]⟩ ./。]

6 Experiments

Evaluation methodology for bracketing is controversial because of varying perspectives on what the "gold standard" should be. We identify two prototypical positions, and give results for both. One position uses a linguistic evaluation criterion, where accuracy is measured against some theoretic notion of constituent structure. The other position uses a *functional* evaluation criterion, where the "correctness" of a bracketing depends on its utility with respect to the application task at hand. For example, here we consider a bracket-pair functionally useful if it correctly identifies phrasal translations—especially where the phrases in the two languages are not compositionally derivable solely from obvious word translations. Notice that in contrast, the linguistic evaluation criterion is insensitive to whether the bracketings of the two sentences match each other in any semantic way, as long as the monolingual bracketings in each sentence are correct. In either case, the *bracket precision* gives the proportion of found brackets that agree with the chosen correctness criterion.

All experiments reported in this paper were performed on sentence-pairs from the HKUST English-Chinese Parallel Bilingual Corpus, which consists of governmental transcripts (Wu 1994). The translation lexicon was automatically learned from the same corpus via statistical sentence alignment (Wu 1994) and statistical Chinese word and collocation extraction (Fung & Wu 1994; Wu & Fung 1994), followed by an EM word-translation learning procedure (Wu & Xia 1994). The translation lexicon contains an English vocabulary of approximately 6,500 words and a Chinese vocabulary of approximately 5,500 words. The mapping is many-to-many, with an average of 2.25 Chinese translations per English word. The translation accuracy is imperfect (about 86% percent weighted precision), which turns out to cause many of the bracketing errors.

Approximately 2,000 sentence-pairs with both English and Chinese lengths of 30 words or less were extracted from our corpus and bracketed using the algorithm described. Several additional criteria were used to filter out unsuitable sentence-pairs. If the lengths of the pair of sentences differed by more than a 2:1 ratio, the pair was rejected; such a difference usually arises as the result of an earlier error in automatic sentence alignment. Sentences containing more than one word absent from the translation lexicon were also rejected; the bracketing method is not intended to be robust against lexicon inadequacies. We also rejected sentence pairs with fewer than two matching words, since this gives the bracketing algorithm no discriminative leverage; such pairs accounted for less than 2% of the input data. A random sample of the bracketed sentence pairs was then drawn, and the bracket precision was computed under each criterion for correctness. Additional examples are shown in Figure 5.

Under the linguistic criterion, the monolingual bracket precision was 80.4% for the English sentences, and 78.4% for the Chinese sentences. Of course, monolingual grammar-based bracketing methods can achieve higher precision, but such tools assume grammar resources that may not be available, such as good Chinese grammars. Moreover, if a good monolingual bracketer is available, its output can easily be incorporated in much the same way as punctuation constraints, thereby combining the best of both worlds. Under the functional criterion, the parallel bracket precision was 72.5%, lower than the monolingual precision since brackets can be correct in one language but not the other. Grammar-based bracketing methods cannot directly produce results of a comparable nature.

7 Conclusion

We have proposed a new tool for the corpus linguist's arsenal: a method for simultaneously bracketing both halves of a parallel bilingual corpus, using only a word translation lexicon. The method can also be seen as a word alignment algorithm that employs a realistic distortion model and aligns consituents as well as words. The basis of the approach is a new *inversion-invariant transduction grammar* formalism.

Various extension strategies for simultaneous segmentation, positional biases, punctuation constraints, singleton rebalancing, and bracket flattening have been introduced. Parallel bracketing exploits a relatively untapped source of constraints, in that parallel bilingual sentences are used to mutually analyze each other. The model nonetheless retains a high degree of compatibility with more conventional monolingual formalisms and methods.

The bracketing and alignment of parallel corpora can be fully automatized with zero initial knowledge resources, with the aid of automatic procedures for learning word translation lexicons. This is particularly valuable for work on languages for which online knowledge resources are relatively scarce compared with English.

Acknowledgement

I would like to thank Xuanyin Xia, Eva Wai-Man Fong, Pascale Fung, and Derick Wood.

References

BLACK, EZRA, ROGER GARSIDE, & GEOFFREY LEECH (eds.). 1993. *Statistically-driven computer grammars of English: The IBM/Lancaster approach*. Amsterdam: Editions Rodopi.

BROWN, PETER F., JOHN COCKE, STEPHEN A. DELLAPIETRA, VINCENT J. DELLAPIETRA, FREDERICK JELINEK, JOHN D. LAFFERTY, ROBERT L. MERCER, & PAUL S. ROOSSIN. 1990. A statistical approach to machine translation. *Computational Linguistics*, 16(2):29–85.

BROWN, PETER F., STEPHEN A. DELLAPIETRA, VINCENT J. DELLAPIETRA, & ROBERT L. MERCER. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311.

CHANG, CHAO-HUANG & CHENG-DER CHEN. 1993. HMM-based part-of-speech tagging for Chinese corpora. In *Proceedings of the Workshop on Very Large Corpora*, 40–47, Columbus, Ohio.

CHIANG, TUNG-HUI, JING-SHIN CHANG, MING-YU LIN, & KEH-YIH SU. 1992. Statistical models for word segmentation and unknown resolution. In *Proceedings of ROCLING-92*, 121–146.

CHURCH, KENNETH W. 1993. Char-align: A program for aligning parallel texts at the character level. In *Proceedings of the 31st Annual Conference of the Association for Computational Linguistics*, 1–8, Columbus, OH.

DAGAN, IDO & KENNETH W. CHURCH. 1994. Termight: Identifying and translating technical terminology. In *Proceedings of the Fourth Conference on Applied Natural Language Processing*, 34–40, Stuttgart.

DAGAN, IDO, KENNETH W. CHURCH, & WILLIAM A. GALE. 1993. Robust bilingual word alignment for machine aided translation. In *Proceedings of the Workshop on Very Large Corpora*, 1–8, Columbus, OH.

FUNG, PASCALE & KENNETH W. CHURCH. 1994. K-vec: A new approach for aligning parallel texts. In *Proceedings of the Fifteenth International Conference on Computational Linguistics*, 1096–1102, Kyoto.

FUNG, PASCALE & KATHLEEN MCKEOWN. 1994. Aligning noisy parallel corpora across language groups: Word pair feature matching by dynamic time warping. In *AMTA-94, Association for Machine Translation in the Americas*, 81–88, Columbia, Maryland.

FUNG, PASCALE & DEKAI WU. 1994. Statistical augmentation of a Chinese machine-readable dictionary. In *Proceedings of the Second Annual Workshop on Very Large Corpora*, 69–85, Kyoto.

GALE, WILLIAM A. & KENNETH W. CHURCH. 1991. A program for aligning sentences in bilingual corpora. In *Proceedings of the 29th Annual Conference of the Association for Computational Linguistics*, 177–184, Berkeley.

GALE, WILLIAM A., KENNETH W. CHURCH, & DAVID YAROWSKY. 1992. Using bilingual materials to develop word sense disambiguation methods. In *Fourth International Conference on Theoretical and Methodological Issues in Machine Translation*, 101–112, Montreal.

LIN, MING-YU, TUNG-HUI CHIANG, & KEH-YIH SU. 1993. A preliminary study on unknown word problem in Chinese word segmentation. In *Proceedings of ROCLING-93*, 119–141.

LIN, YI-CHUNG, TUNG-HUI CHIANG, & KEH-YIH SU. 1992. Discrimination oriented probabilistic tagging. In *Proceedings of ROCLING-92*, 85–96.

MAGERMAN, DAVID M. & MITCHELL P. MARCUS. 1990. Parsing a natural language using mutual information statistics. In *Proceedings of AAAI-90, Eighth National Conference on Artificial Intelligence*, 984–989.

PEREIRA, FERNANDO & YVES SCHABES. 1992. Inside-outside reestimation from partially bracketed corpora. In *Proceedings of the 30th Annual Conference of the Association for Computational Linguistics*, 128–135, Newark, DE.

SPROAT, RICHARD, CHILIN SHIH, WILLIAM GALE, & N. CHANG. 1994. A stochastic word segmentation algorithm for a Mandarin text-to-speech system. In *Proceedings of the 32nd Annual Conference of the Association for Computational Linguistics*, Las Cruces, New Mexico. To appear.

VITERBI, ANDREW J. 1967. Error bounds for convolutional codes and an asymptotically optimal decoding algorithm. *IEEE Transactions on Information Theory*, 13:260–269.

WU, DEKAI. 1994. Aligning a parallel English-Chinese corpus statistically with lexical criteria. In *Proceedings of the 32nd Annual Conference of the Association for Computational Linguistics*, 80–87, Las Cruces, New Mexico.

WU, DEKAI, 1995. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. In preparation.

WU, DEKAI & PASCALE FUNG. 1994. Improving Chinese tokenization with linguistic filters on statistical lexical acquisition. In *Proceedings of the Fourth Conference on Applied Natural Language Processing*, 180–181, Stuttgart.

WU, DEKAI & XUANYIN XIA. 1994. Learning an English-Chinese lexicon from a parallel corpus. In *AMTA-94, Association for Machine Translation in the Americas*, 206–213, Columbia, Maryland.

WU, ZIMIN & GWYNETH TSENG. 1993. Chinese text segmentation for text retrieval: Achievements and problems. *Journal of The American Society for Information Science*, 44(9):532–542.