To appear in *IJCAI-95, Proc. of the 14th International Joint Conference on Artificial Intelligence*, Montreal: Aug. 95.

Stochastic Inversion Transduction Grammars, with Application to Segmentation, Bracketing, and Alignment of Parallel Corpora

Dekai Wu[*]

Technical Report HKUST-CS95-30
June 1995

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

Abstract

We introduce (1) a novel *stochastic inversion transduction grammar* formalism for bilingual language modeling of sentence-pairs, and (2) the concept of *bilingual parsing* with potential application to a variety of parallel corpus analysis problems. The formalism combines three tactics against the constraints that render finite-state transducers less useful: it skips directly to a context-free rather than finite-state base, it permits a minimal extra degree of ordering flexibility, and its probabilistic formulation admits an efficient maximum-likelihood bilingual parsing algorithm. A convenient normal form is shown to exist, and we discuss a number of examples of how stochastic inversion transduction grammars bring bilingual constraints to bear upon problematic corpus analysis tasks.

Stochastic Inversion Transduction Grammars, with Application to Segmentation, Bracketing, and Alignment of Parallel Corpora

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

Abstract

We introduce (1) a novel *stochastic inversion transduction grammar* formalism for bilingual language modeling of sentence-pairs, and (2) the concept of *bilingual parsing* with potential application to a variety of parallel corpus analysis problems. The formalism combines three tactics against the constraints that render finite-state transducers less useful: it skips directly to a context-free rather than finite-state base, it permits a minimal extra degree of ordering flexibility, and its probabilistic formulation admits an efficient maximum-likelihood bilingual parsing algorithm. A convenient normal form is shown to exist, and we discuss a number of examples of how stochastic inversion transduction grammars bring bilingual constraints to bear upon problematic corpus analysis tasks.

1 Introduction

We introduce a general formalism for modeling of bilingual sentence pairs, known as an *inversion transduction grammar*, with potential application in a variety of corpus analysis areas. Transducer models, especially of the finite-state family, have long been known. However, finite-state transducers impose identical ordering constraints upon both streams, confining their applicability to NLP tasks to narrowly restricted domains outside of which transduction has received relatively little attention. The inversion transduction grammar formalism skips directly to a context-free rather than finite-state base and permits one extra degree of ordering flexibility, while retaining properties necessary for efficient computation, thereby sidestepping the limitations of traditional transducers.

In tandem with the concept of bilingual language modeling, we propose the concept of bilingual parsing, where the input is a sentence-*pair* rather than a sentence. Though inversion transduction grammars remain inadequate as full-fledged translation models, bilingual parsing with simple inversion transduction grammars turns out to be very useful for parallel corpus analysis when the true grammar is not fully known. Parallel bilingual corpora have been shown to provide a 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; Fung & Church 1994; Wu & Xia 1994; Fung & McKeown 1994). The primary purpose of bilingual parsing with inversion transduction grammars is not to flag ungrammatical inputs; rather the aim is to extract structure from the input data which is assumed to be grammatical, in kindred spirit with robust parsing. Sample applications to segmentation, bracketing, phrasal alignment, and parsing are surveyed later in this paper. The formalism's uniform integration of various types of bracketing and alignment constraints is one of its chief strengths.

We begin below by laying out the basic formalism, then show that reduction to a normal form is possible. Afterwards we introduce a stochastic version and give an algorithm for finding the optimal bilingual parse of a sentence-pair. The formalism is independent of the languages; we give examples and applications using Chinese and English, because languages from different families provide a more rigorous testing ground.

2 Inversion Transduction Grammars

As a stepping stone to inversion transduction grammarss, we first consider what a symmetric context-free transduction grammar (CFTG) would look like. The utility of finite-state transducers is well-known for narrow tasks such as nominal, number, and temporal phrase normalization, text-to-speech conversion, and analysis of inflectional morphology (Gazdar & Mellish 1989), but for general corpus analysis FSTs are inadequate.

By *transduction* we mean that two output streams are generated, one for each language. (Transducers are often presented as having one input and one output stream, but this view works better for deterministic finite-state machines than for the non-deterministic models we are using here. Moreover for our application the two languages' role is symmetric.) In a CFTG, every terminal symbol is marked for a particular output stream. Thus, each rewrite rule emits not one but two streams. For example, a rewrite rule of the form $A \rightarrow Bx_1y_2Cz_1$ means that the terminal symbols x and z are symbols of the language L_1 emitted on stream 1, while y is a symbol of the language L_2 emitted on stream 2. It follows that every nonterminal stands for a class of derivable substring *pairs*.

We can use a CFTG to model the generation of bilingual sentence pairs. As a mnemonic convention, we usually use the alternative notation $A \rightarrow B\ x/y\ C\ z/\epsilon$ to associate matching output tokens. Though this additional information has no formal generative effect, it reminds us that x/y must be a

(a)	S	$\rightarrow$	[SP Stop]
	SP	$\rightarrow$	[NP VP] | [NP VV] | [NP V]
	PP	$\rightarrow$	[Prep NP]
	NP	$\rightarrow$	[Det NN] | [Det N] | [Pro] | [NP Conj NP]
	NN	$\rightarrow$	[A N] | [NN PP]
	VP	$\rightarrow$	[Aux VP] | [Aux VV] | [VV PP]
	VV	$\rightarrow$	[V NP] | [Cop A]
	Det	$\rightarrow$	the/ϵ
	Prep	$\rightarrow$	to/向
	Pro	$\rightarrow$	I/我 | you/你
	N	$\rightarrow$	authority/管理局 | secretary/司
	A	$\rightarrow$	accountable/負責 | financial/財政
	Conj	$\rightarrow$	and/和
	Aux	$\rightarrow$	will/將會
	Cop	$\rightarrow$	be/ϵ
	Stop	$\rightarrow$	./。
(b)	VP	$\rightarrow$	$\langle$VV PP$\rangle$

Figure 1: *(a)* A context-free transduction grammar. *(b)* An inverted-orientation production.

valid entry in the translation lexicon. We call a matched terminal symbol pair such as x/y a *couple*. The null symbol ϵ means that no output token is generated. We call x/ϵ an L_1-*singleton*, and ϵ/y an L_2-*singleton*.

Consider the simple context-free transduction grammar fragment shown in Figure 1(a). (It will become apparent below why we explicitly include brackets around right-hand sides containing nonterminals, which are usually omitted with standard CFGs.) The transduction grammar can generate, for instance, the following pair of English and Chinese sentences in translation:

(1) a. [[[[The [Financial Secretary]$_{NN}$]$_{NP}$ and [I]$_{NP}$]$_{NP}$ [will [be accountable]$_{VV}$]$_{VP}$]$_{SP}$ ./。]$_S$

b. [[[[[財政 司]$_{NN}$]$_{NP}$ 和 [我]$_{NP}$]$_{NP}$ [將會 [負責]$_{VV}$]$_{VP}$]$_{SP}$ ./。]$_S$

Notice that each nonterminal derives two substrings, one in each language. The two substrings are counterparts of each other. In fact, it is natural to write the parse trees together:

(2) [[[[The/ϵ [Financial/財政 Secretary/司]$_{NN}$]$_{NP}$ and/和 [I/我]$_{NP}$]$_{NP}$ [will/將會 [be/ϵ accountable/負責]$_{VV}$]$_{VP}$]$_{SP}$ ./。]$_S$

Of course, in general context-free transduction grammars are not very useful, precisely because they require the two languages to share exactly the same grammatical structure (modulo those distinctions that can be handled with lexical singletons). For example, the following sentence pair from our corpus cannot be generated:

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

b. 管理局將會向財政司負責。

To make transduction grammars truly useful for bilingual tasks, we must escape the rigid parallel ordering constraint of context-free transduction grammars. At the same time, any relaxation of constraints must be traded off against increases in the computational complexity of parsing, which may easily become exponential. The key is to make the relaxation relatively modest but still handle a wide range of ordering variations.

The inversion transduction grammar (ITG) formalism only minimally extends the generative power of a context-free transduction grammar,[1] yet turns out to be surprisingly effective. The productions of an inversion transduction grammar are interpreted just as in a context-free transduction grammar, except that two possible *orientations* are allowed. Pure context-free transduction grammars have the implicit characteristic that for both output streams, the symbols generated by the right-hand side constituents of a production are concatenated in the same left-to-right order. Inversion transduction grammars also allow such productions, which are said to have *straight* orientation. In addition, however, inversion transduction grammars allow productions with *inverted* orientation, which generate output for stream 2 by emitting the constituents on a production's right-hand side in *right-to-left* order. We indicate a production's orientation with explicit notation for the two varieties of concatenation operators on string-pairs. 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 $C_2 = B_2A_2$. Since inversion is permitted at any level of rule expansion, a derivation may intermix productions of either orientation within the parse tree. For example, if the inverted-orientation production of Figure 1(b) is added to the earlier context-free transduction grammar, sentence-pair (3) can then be generated as follows:

(4) a. [[[The Authority]$_{NP}$ [will [[be accountable]$_{VV}$ [to [the [[Financial Secretary]$_{NN}$]$_{NNN}$]$_{NP}$]$_{PP}$]$_{VP}$]$_{VP}$]$_{SP}$ ./。]$_S$

b. [[[管理局]$_{NP}$ [將會 [[向 [[[財政 司]$_{NN}$]$_{NNN}$]$_{NP}$]$_{PP}$ [負責]$_{VV}$]$_{VP}$]$_{VP}$]$_{SP}$ ./。]$_S$

We can show the common structure of the two sentences more clearly and compactly with the aid of the $\langle\rangle$ notation:

(5) [[[The/ϵ Authority/管理局]$_{NP}$ [will/將會 $\langle$[be/ϵ accountable/負責]$_{VV}$ [to/向 [the/ϵ [[Financial/財政 Secretary/司]$_{NN}$]$_{NNN}$]$_{NP}$]$_{PP}$ $\rangle_{VP}$]$_{VP}$]$_{SP}$ ./。]$_S$

Alternatively, a graphical parse tree notation is shown in Figure 2, where the $\langle\rangle$ level of bracketing is indicated by a 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.

Parsing, in the case of an ITG, means building matched constituents for input 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 yields labelled bracketings for both sentences, as well as a bracket alignment indicating the parallel constituents between the sentences. The constituent alignment includes a word alignment as a byproduct.

[1] The expressiveness of CFTGs is equivalent to pushdown transducers (Savitch 1982). ITGs are of greater expressiveness and can be seen in terms of syntax-directed transduction (Lewis & Stearns 1968) but this view is too general to be of much help. Also note that parsing is bilingual with ITGs, whereas the pushdown and syntax-directed transduction frameworks are designed for monolingual parsing in tandem with generation.

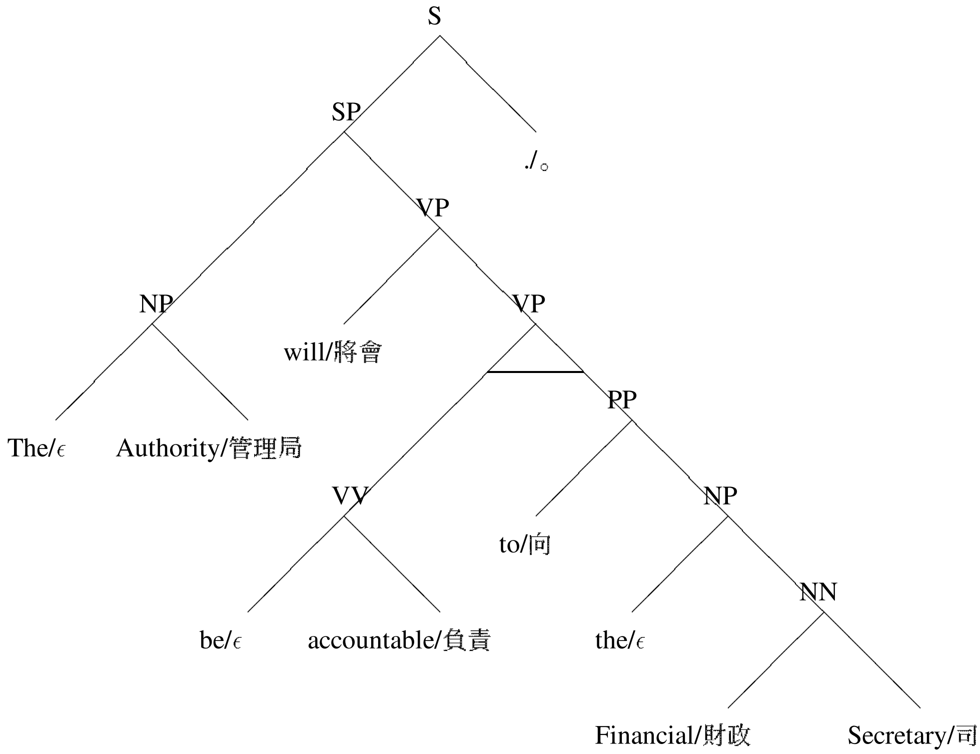

Figure 2: Inversion transduction parse tree.

Clearly, the nonterminals of an ITG must be chosen in a somewhat different manner than for a monolingual grammar, since they must simultaneously account for syntactic patterns of both languages. Moreover, certain phenomena with underlying structure that is not context-free—particularly, ellipsis and coordination—fall outside the expressiveness of ITGs if the surface structures of the two languages do not parallel each other. Nevertheless, a wide range of ordering variation between the languages can be accommodated by appropriate decomposition of productions (and thus constituents), in conjuction with introduction of new auxiliary nonterminals where needed. In fact ITGs can generate all 34 possible alignments between subsequences of length 3, and 207 out of the 209 possible alignments between length 4 subsequences.[2] Messy alignments such as that in Figure 3 can be handled by interleaving orientations:

(6) [⟨⟨Where/那裡 is/在⟩ [[the/ϵ ⟨Secretary/司 [of/ϵ Finance/財政]⟩] ⟨when/時 needed/有需要⟩]⟩ ?/?]

We stress again that the primary purpose of ITGs is robust analysis rather than grammaticality determination, and therefore writing grammars is made much easier since the grammars can be minimal and very leaky. We consider elsewhere an extreme special case of leaky ITGs, *inversion-invariant transduction grammars*, in which all productions occur with both orientations (Wu 1995a). As the applications below demonstrate, the bilingual lexical constraints carry greater importance than the tightness of the grammar.

[2] See also Section 7. The analysis of ordering flexibility is omitted for space reasons, but is given elsewhere (Wu 1995b).

3 A Normal Form for ITGs

We show here that every inversion transduction grammar can be expressed as an equivalent grammar in a normal form that simplifies algorithms and analyses on ITGs. In particular, the parsing algorithm of the next section operates on ITGs in normal form. The lemmas' proofs are omitted.

Lemma 1 *For any inversion transduction grammar G, there exists an equivalent inversion 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. *else G' contains no productions of the form $A \rightarrow \epsilon/\epsilon$.*

Lemma 2 *For any inversion transduction grammar G, there exists an equivalent inversion 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 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 $A \rightarrow B$.*

Theorem 1 *For any inversion transduction grammar G, there exists an equivalent inversion transduction grammar G' in which every production takes one of the following forms:*

$$\begin{array}{llllll} S \rightarrow \epsilon/\epsilon & A \rightarrow x/\epsilon & A \rightarrow [B\,C] \\ A \rightarrow x/y & A \rightarrow \epsilon/y & A \rightarrow \langle B\,C \rangle \end{array}$$

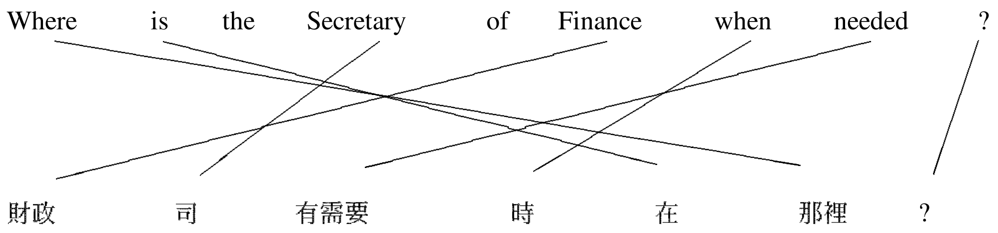

Figure 3: A permissible complex alignment.

Proof By Lemmas 1, 2, and 3, we may assume G contains only productions of the form $S \to \epsilon/\epsilon$, $A \to x/y$, $A \to x/\epsilon$, $A \to \epsilon/y$, $A \to [B_1 B_2]$, $A \to \langle B_1 B_2 \rangle$, $A \to [B_1 \dots B_n]$, and $A \to \langle B_1 \dots B_n \rangle$ where $n \geq 3$. Include in G' all productions of the first six types. The remaining two types are transformed as follows.

For each production of the form $A \to [B_1 \dots B_n]$ we introduce new nonterminals $X_1 \dots X_{n-2}$ in order to replace the production with the set of rules $A \to [B_1 X_1], X_1 \to [B_2 X_2], \dots, X_{n-3} \to [B_{n-2} X_{n-2}], X_{n-2} \to [B_{n-1} B_n]$. Let (E, C) be any string-pair derivable from $A \to [B_1 \dots B_n]$, where E is output on stream 1 and C on stream 2. Define E^i as the substring of E derived from B_i, and similarly define C^i. Then X_i generates $(E^{i+1} \cdots E^n, C^{i+1} \cdots C^n)$ for all $1 \leq i < n$, so the new production $A \to [B_1 X_1]$ also generates (E, C). No additional string-pairs are generated due to the new productions (since each X_i is only reachable from X_{i-1} and X_1 is only reachable from A).

For each production of the form $A \to \langle B_1 \dots B_n \rangle$ we replace the production with the set of rules $A \to \langle B_1 X_1 \rangle, X_1 \to \langle B_2 X_2 \rangle, \dots, X_{n-3} \to \langle B_{n-2} X_{n-2} \rangle, X_{n-2} \to \langle B_{n-1} B_n \rangle$. Let (E, C) be any string-pair derivable from $A \to \langle B_1 \dots B_n \rangle$, where E is output on stream 1 and C on stream 2. Again define E^i and C^i as the substrings derived from the B_i, but in this case $(E, C) = (E^1 \cdots E^n, C^n \cdots C^1)$. Then Y_i generates $(E^{i+1} \dots E^n, C^{n-i+1} \dots C^1)$ for all $1 \leq i < n$, so the new production $A \to \langle B_1 Y_1 \rangle$ also generates (E, C). Again no additional string-pairs are generated due to the new productions. $\square$

Henceforth all transduction grammars will be assumed to be in normal form.

4 Stochastic Inversion Transduction Grammars

In a stochastic ITG (SITG), a probability is associated with each rewrite rule. For example, the probability of the rule NN $\stackrel{0.4}{\to}$ [A N] is $a_{\text{NN}\to\text{[A N]}} = 0.4$. The probability of a lexical rule A $\stackrel{0.001}{\to} x/y$ is $b_{\text{A}}(x, y) = 0.001$. Let W_1, W_2 be the vocabulary sizes of the two languages, and $\mathcal{N}$ be the set of nonterminals with indices $1, \dots, N$. Then for every $1 \leq i \leq N$, the production probabilities are subject to the constraint that

$$\sum_{1 \leq j,k \leq N} (a_{i\to[jk]} + a_{i\to\langle jk \rangle}) + \sum_{\substack{1 \leq x \leq W_1 \\ 1 \leq y \leq W_2}} b_i(x, y) = 1$$

We now introduce an algorithm for parsing with stochastic ITGs, that computes an optimal parse given a sentence-pair using dynamic programming (DP). 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) and to chart parsers (Earley 1970).

Let the input English sentence be $\mathbf{e}_1, \dots, \mathbf{e}_T$ and the corresponding input Chinese sentence be $\mathbf{c}_1, \dots, \mathbf{c}_V$. As an abbreviation we write $\mathbf{e}_{s..t}$ for the sequence of words $\mathbf{e}_{s+1}, \mathbf{e}_{s+2}, \dots, \mathbf{e}_t$, and similarly for $\mathbf{c}_{u..v}$. It is convenient to use a 4-tuple of the form $q = (s, t, u, v)$ to identify each node of the parse tree, where the substrings $\mathbf{e}_{s..t}$ and $\mathbf{c}_{u..v}$ both derive from the node q. Denote the nonterminal label on q by $\ell(q)$. Then for any node $q = (s, t, u, v)$, define

$$\delta_q(i) = \delta_{stuv}(i) = \max_{\text{subtrees of } q} P[\text{subtree of } q, \ell(q) = i, i \stackrel{*}{\Rightarrow} \mathbf{e}_{s..t}/\mathbf{c}_{u..v}]$$

as the maximum probability of any derivation from i that successfully parses both $\mathbf{e}_{s..t}$ and $\mathbf{c}_{u..v}$. Then the best parse of the sentence pair has probability $\delta_{0,T,0,V}(\text{S})$.

The algorithm computes $\delta_{0,T,0,V}(\text{S})$ using the following recurrences. Note that we generalize argmax to the case where maximization ranges over multiple indices, by making it vector-valued. Also note that $[\,]$ and $\langle\rangle$ are simply constants, written mnemonically. The condition $(S - s)(t - S) + (U - u)(v - U) \neq 0$ is a way to specify that the substring in one but not both languages may be split into an empty string ϵ and the substring itself; this ensures that the recursion terminates, but permits words that have no match in the other language to map to an ϵ instead.

1. Initialization

$$\delta_{t-1,t,v-1,v}(i) = b_i(\mathbf{e}_t/\mathbf{c}_v), \qquad \begin{matrix} 1 \leq t \leq T \\ 1 \leq v \leq V \end{matrix}$$

$$\delta_{t-1,t,v,v}(i) = b_i(\mathbf{e}_t/\epsilon), \qquad \begin{matrix} 1 \leq t \leq T \\ 0 \leq v \leq V \end{matrix}$$

$$\delta_{t,t,v-1,v}(i) = b_i(\epsilon/\mathbf{c}_v), \qquad \begin{matrix} 0 \leq t \leq T \\ 1 \leq v \leq V \end{matrix}$$

2. Recursion For all i, s, t, u, v such that $\begin{cases} 1 \leq i \leq N \\ 0 \leq s < t \leq T \\ 0 \leq u < v \leq V \\ t - s + v - u > 2 \end{cases}$

$$\delta_{stuv}(i) = \max[\delta^{[]}_{stuv}(i), \delta^{\langle\rangle}_{stuv}(i)]$$

$$\theta_{stuv}(i) = \begin{cases} [\,] & \text{if } \delta^{[]}_{stuv}(i) \geq \delta^{\langle\rangle}_{stuv}(i) \\ \langle\rangle & \text{otherwise} \end{cases}$$

where

$$\delta^{[]}_{stuv}(i) = \max_{\substack{1\le j\le N\\ 1\le k\le N\\ s\le S\le t\\ u\le U\le v\\ (S-s)(t-S)+(U-u)(v-U)\ne 0}} a_{i\to[jk]}\,\delta_{sSuU}(j)\,\delta_{StUv}(k)$$

$$\begin{bmatrix} \iota^{[]}_{stuv}(i)\\ \kappa^{[]}_{stuv}(i)\\ \sigma^{[]}_{stuv}(i)\\ \upsilon^{[]}_{stuv}(i)\end{bmatrix} = \operatorname*{argmax}_{\substack{1\le j\le N\\ 1\le k\le N\\ s\le S\le t\\ u\le U\le v\\ (S-s)(t-S)+(U-u)(v-U)\ne 0}} a_{i\to[jk]}\,\delta_{sSuU}(j)\,\delta_{StUv}(k)$$

$$\delta^{\langle\rangle}_{stuv}(i) = \max_{\substack{1\le j\le N\\ 1\le k\le N\\ s\le S\le t\\ u\le U\le v\\ (S-s)(t-S)+(U-u)(v-U)\ne 0}} a_{i\to\langle jk\rangle}\,\delta_{sSUv}(j)\,\delta_{StuU}(k)$$

$$\begin{bmatrix} \iota^{\langle\rangle}_{stuv}(i)\\ \kappa^{\langle\rangle}_{stuv}(i)\\ \sigma^{\langle\rangle}_{stuv}(i)\\ \upsilon^{\langle\rangle}_{stuv}(i)\end{bmatrix} = \operatorname*{argmax}_{\substack{1\le j\le N\\ 1\le k\le N\\ s\le S\le t\\ u\le U\le v\\ (S-s)(t-S)+(U-u)(v-U)\ne 0}} a_{i\to\langle jk\rangle}\,\delta_{sSUv}(j)\,\delta_{StuU}(k)$$

3. Reconstruction Initialize by setting the root of the parse tree to $q_1 = (0, T, 0, V)$ and its nonterminal label to $\ell(q_1) =$ S. The remaining descendants in the optimal parse tree are then given recursively for any $q = (s, t, u, v)$ by:

$$\text{LEFT}(q) = \begin{cases} \text{NIL} & \text{if } t-s+v-u\le 2\\ (s, \sigma^{[]}_q(\ell(q)), u, \upsilon^{[]}_q(\ell(q))) & \text{if } \theta_q(\ell(q)) = [\,]\\ (s, \sigma^{\langle\rangle}_q(\ell(q)), \upsilon^{\langle\rangle}_q(\ell(q))) & \text{if } \theta_q(\ell(q)) = \langle\rangle \end{cases}$$

$$\text{RIGHT}(q) = \begin{cases} \text{NIL} & \text{if } t-s+v-u\le 2\\ (\sigma^{[]}_q(\ell(q)), t, \upsilon^{[]}_q(\ell(q)), v) & \text{if } \theta_q(\ell(q)) = [\,]\\ (\sigma^{\langle\rangle}_q(\ell(q)), t, u, \upsilon^{\langle\rangle}_q(\ell(q))) & \text{if } \theta_q(\ell(q)) = \langle\rangle \end{cases}$$

$$\ell(\text{LEFT}(q)) = \iota_q^{\theta_q(\ell(q))}(\ell(q))$$

$$\ell(\text{RIGHT}(q)) = \kappa_q^{\theta_q(\ell(q))}(\ell(q))$$

The time complexity of this algorithm in the general case is $\Theta(N^3T^3V^3)$ where N is the number of distinct nonterminals and T and V are the lengths of the two sentences. This is a factor of V^3 more than monolingual chart parsing, but has turned out to remain quite practical for corpus analysis, where parsing need not be real-time.

5 Applications: Translation-Driven Segmentation

Segmentation of the input sentences is an important step in preparing bilingual corpora for various learning procedures. Different languages realize the same concept using varying numbers of words; a single English word may surface as a compound in French. This complicates the problem of matching the words between a sentence pair, since it means that compounds or collocations must sometimes be treated as lexical units. The translation lexicon is assumed to contain collocation translations to facilitate such multi-word matchings. However, the input sentences do not come broken into appropriately matching chunks, so it is up to the parser to decide when to break up potential collocations into individual words.

The problem is particularly acute for English and Chinese because word boundaries are not orthographically marked in Chinese text, so not even a default chunking exists, upon which word matchings could be postulated. (Sentences (2) and (2) demonstrate why the obvious trick of taking single characters as words is not a workable strategy.) The usual Chinese NLP architecture first preprocesses input text through a word segmentation module (Chiang *et al.* 1992; Chang & Chen 1993; Lin *et al.* 1993; Wu & Tseng 1993; Sproat *et al.* 1994; Wu & Fung 1994), but clearly bilingual parsing will be hampered by any errors arising from segmentation ambiguities that could not be resolved in the isolated monolingual context because even if the Chinese segmentation is acceptable monolingually, it may not agree with the words present in the English sentence. Matters are made still worse by unpredictable omissions in the translation lexicon, even for valid compounds.

We therefore extend the algorithm to optimize the Chinese sentence segmentation in conjunction with the bracketing process. Note that the notion of a Chinese "word" is a longstanding linguistic question, that our present notion of segmentation does not address. We adhere here to a purely task-driven definition of what a correct "segmentation" is, namely that longer segments are desirable only when no compositional translation is possible. The algorithm is modified to include the following computations, and remains the same otherwise:

1. Initialization

$$\delta^0_{stuv}(i) = b_i(\mathbf{e}_{s..t}/\mathbf{c}_{u..v}), \qquad \begin{matrix} 0\le s\le t\le T\\ 0\le u\le v\le V\end{matrix}$$

2. Recursion

$$\delta_{stuv}(i) = \max[\delta^{[]}_{stuv}(i), \delta^{\langle\rangle}_{stuv}(i), \delta^{0}_{stuv}(i)]$$

$$\theta_{stuv}(i) = \begin{cases} [\,] & \text{if } \delta^{[]}_{stuv}(i) > \delta^{\langle\rangle}_{stuv}(i)\\ & \quad\text{and } \delta^{[]}_{stuv}(i) > \delta^{0}_{stuv}(i)\\ \langle\rangle & \text{if } \delta^{\langle\rangle}_{stuv}(i) > \delta^{[]}_{stuv}(i)\\ & \quad\text{and } \delta^{\langle\rangle}_{stuv}(i) > \delta^{0}_{stuv}(i)\\ 0 & \text{otherwise}\end{cases}$$

3. Reconstruction

$$\text{LEFT}(q) = \begin{cases} \text{NIL} & \text{if } t-s+v-u\le 2\\ (s, \sigma^{[]}_q(\ell(q)), u, \upsilon^{[]}_q(\ell(q))) & \text{if } \theta_q(\ell(q)) = [\,]\\ (s, \sigma^{\langle\rangle}_q(\ell(q)), \upsilon^{\langle\rangle}_q(\ell(q))) & \text{if } \theta_q(\ell(q)) = \langle\rangle\\ \text{NIL} & \text{otherwise} \end{cases}$$

$$\text{RIGHT}(q) = \begin{cases} \text{NIL} & \text{if } t-s+v-u\le 2\\ (\sigma^{[]}_q(\ell(q)), t, \upsilon^{[]}_q(\ell(q)), v) & \text{if } \theta_q(\ell(q)) = [\,]\\ (\sigma^{\langle\rangle}_q(\ell(q)), t, u, \upsilon^{\langle\rangle}_q(\ell(q))) & \text{if } \theta_q(\ell(q)) = \langle\rangle\\ \text{NIL} & \text{otherwise} \end{cases}$$

In our experience this method has proven extremely effective for avoiding missegmentation pitfalls, essentially erring only in pathological cases involving coordination constructions or lexicon coverage inadequacies. The method is also straightforward to employ in tandem with other applications such as those below.

6 Applications: Bracketing

Bracketing is another intermediate corpus annotation, useful especially when a full-coverage grammar with which to parse a corpus is unavailable (for Chinese, an even more common situation than with English). 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).

Our method based on SITGs operates on the novel principle that lexical correspondences between parallel sentences yields information from which partial bracketings for both sentences can be extracted. The assumption that no grammar is available means that constituent categories are not differentiated. Instead, a generic *bracketing inversion transduction grammar* is employed, containing only one nonterminal symbol, A, which rewrites either recursively as a pair of A's or as a single terminal-pair:

$$\begin{array}{lll} \mathrm{A} \xrightarrow{a} [\mathrm{A\ A}] & & \\ \mathrm{A} \xrightarrow{a} \langle \mathrm{A\ A} \rangle & & \\ \mathrm{A} \xrightarrow{b_{ij}} u_i/v_j & \text{for all } i, j \text{ lexical translations} \\ \mathrm{A} \xrightarrow{b_{i\epsilon}} u_i/\epsilon & \text{for all } i \text{ English vocabulary} \\ \mathrm{A} \xrightarrow{b_{\epsilon j}} \epsilon/v_j & \text{for all } j \text{ Chinese vocabulary} \end{array}$$

Longer productions with fanout > 2 are not needed; we show in (Wu 1995a) that this minimal transduction grammar in normal form is generatively equivalent to any reasonable ITG for bracketing. Moreover, we also show how postprocessing using rotation and flattening operations restores the fanout flexibility so that an output bracketing can hold more than two immediate constituents, as shown in Figure 4.

The b_{ij} distribution actually encodes the English-Chinese translation lexicon. We have been using a lexicon that was automatically learned from the HKUST English-Chinese Parallel Bilingual 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 latter stage gives us the b_{ij} probabilities directly. For the two singleton productions, which permit any word in either sentence to be unmatched, a small ϵ-constant can be chosen for the probabilities $b_{i\epsilon}$ and $b_{\epsilon j}$, so that the optimal bracketing resorts to these productions only when it is otherwise impossible to match words.

An experiment was carried out as follows. 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. Examples are shown in Figure 4.

The bracket precision was 80.4% for the English sentences, and 78.4% for the Chinese sentences, as judged against manual bracketings. Inspection showed the errors to be due largely to imperfections of our translation lexicon, which contains approximately 6,500 English words and 5,500 Chinese words with about 86% translation accuracy (Wu & Xia 1994), so a better lexicon should yield substantial performance improvement. Moreover, if the resources for a good monolingual part-of-speech or grammar-based bracketer such as that of Magerman & Marcus (1990) are available, its output can readily be incorporated in complementary fashion as discussed in Section 8.

7 Applications: Phrasal Alignment

Phrasal Alignment The parsing algorithm can be used to identify phrasal translations within sentence pairs. This is useful especially where the phrases in the two languages are not compositionally derivable solely from obvious word translations, such as [have acquired/ϵ ϵ/學到 new/新 skills/技能] in Figure 4. This principle applies to nested structures also, such as ⟨[ϵ/的工 who/人] [have acquired/ϵ ϵ/學到 new/新 skills/技能]⟩, on up to the sentence level. These examples were found using the minimal bracketing transduction grammar, a relatively weak strategy; we evaluated the precision at 72.5%, which is useful but rather low. Higher precision could be achieved without great effort by employing a small number of broad nonterminal categories.

Word Alignment Under the ITG model, word alignment becomes simply the special case of phrasal alignment at the parse tree leaves. However, this gives us an interesting alternative perspective, from the standpoint of algorithms that match the words between parallel sentences. By themselves word alignments are of little use, but they provide potential anchor points for other applications, or for subsequent learning stages to acquire more interesting structures.

Word alignment is difficult because correct matchings are not usually linearly ordered, i.e., there are crossings. Without some additional constraints, any word position in the source sentence can be matched to any position in the target sentence, an assumption which leads to high error rates. More sophisticated word alignment algorithms therefore attempt to model the intuition that proximate constituents in close relationships in one language remain proximate in the other. The later IBM models are formulated to prefer collocations (Brown *et al.* 1993). In the case of *word_align* (Dagan *et al.* 1993; Dagan & Church 1994), a penalty is imposed according to the deviation from an ideal matching, as constructed by linear

[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 4: Bracketing output examples. (<> = unrecognized input token.)

interpolation.[3]

From this point of view, the proposed technique is a word alignment method that imposes a more realistic distortion penalty. The tree structure reflects the assumption that crossings should not be penalized as long as they are consistent with constituent structure. Figure 5 gives theoretical upper bounds on the matching flexibility as the lengths of the sequences increase, where the constituent structure constraints are reflected by high flexibility up to $m = 4$ and a rapid dropoff thereafter.[3] In other words, ITGs appeal to a language universals hypothesis, that the core arguments of frames, which exhibit great ordering variation between languages, are relatively few and surface in syntactic proximity. Of course this assumption over-simplistically blends syntactic and semantic notions. That semantic frames for different languages share common core arguments is more plausible than syntactic frames. In effect we are relying on the tendency of syntactic arguments to correlate closely with semantics. If in particular cases this assumption does not hold, however, the damage is not too great in that the model will simply drop the offending word matchings (dropping as few as possible).

In experiments with the minimal bracketing transduction grammar, the large majority of errors in word alignment were caused by two outside factors. First, word matchings can be overlooked simply due to deficiencies in our translation lexicon. This accounted for approximately 42% of the errors. Second, sentences containing non-literal translations obviously cannot be aligned down to the word level. This accounted for another approximate 50% of the errors. Excluding these two types of errors, accuracy on word alignment was 96.3%. In other words, the tree-structure constraint is strong enough to prevent most false matches, but almost never inhibits correct word matches when they exist.

8 Applications: Bilingual Constraint Transfer

Monolingual Parse Tree A parse may be available for one of the languages, especially for well-studied languages such as English. Since this eliminates all degrees of freedom in the English sentence structure, the parse of the Chinese sentence must conform with that given for the English. Knowledge of English bracketing is thus used to help parse the Chinese sentence; this method facilitates a kind of transfer of grammatical expertise in one language toward bootstrapping grammar acquisition in another.

A parsing algorithm for this case can be implemented very efficiently. Note that the English parse tree already determines the split point S for breaking $\mathbf{e}_{0..T}$ into two constituent subtrees deriving $\mathbf{e}_{0..S}$ and $\mathbf{e}_{S..T}$ respectively, as well as the nonterminal labels j and k for each subtree. The same then applies recursively to each subtree. We indicate this by turning S, j, and k into deterministic functions on the English constituents, writing S_{st}, j_{st}, and k_{st} to denote the split point and the subtree labels for any constituent $\mathbf{e}_{s..t}$. The following simplifications can then be made to the parsing algorithm:

2. Recursion For all English constituents $\mathbf{e}_{s..t}$ and all i, u, v such that $\left\{ \begin{smallmatrix} 1 \le i \le N \\ 0 \le u < v \le V \end{smallmatrix} \right.$

$$\delta^{[]}_{stuv}(i) = \max_{u \le U \le v} a_{i \to [j_{st} k_{st}]} \, \delta_{s,S_{st},u,U}(j_{st}) \, \delta_{S_{st},t,U,v}(k_{st})$$

$$\upsilon^{[]}_{stuv}(i) = \operatorname*{argmax}_{u \le U \le v} \delta_{s,S_{st},u,U}(j_{st}) \, \delta_{S_{st},t,U,v}(k_{st})$$

$$\delta^{\langle\rangle}_{stuv}(i) = \max_{u \le U \le v} a_{i \to \langle j_{st} k_{st} \rangle} \, \delta_{s,S_{st},U,v}(j_{st}) \, \delta_{S_{st},t,u,U}(k_{st})$$

$$\upsilon^{\langle\rangle}_{stuv}(i) = \operatorname*{argmax}_{u \le U \le v} \delta_{s,S_{st},U,v}(j_{st}) \, \delta_{S_{st},t,u,U}(k_{st})$$

3. Reconstruction

$$\begin{aligned}
\text{LEFT}(q) &= \begin{cases} (s, S_{st}, u, \upsilon^{[]}_q(\ell(q))) & \text{if } \theta_q(\ell(q)) = [\,] \\ (s, S_{st}, \upsilon^{\langle\rangle}_q(\ell(q))) & \text{if } \theta_q(\ell(q)) = \langle\rangle \end{cases} \\
\text{RIGHT}(q) &= \begin{cases} (S_{st}, t, \upsilon^{[]}_q(\ell(q)), v) & \text{if } \theta_q(\ell(q)) = [\,] \\ (S_{st}, t, u, \upsilon^{\langle\rangle}_q(\ell(q))) & \text{if } \theta_q(\ell(q)) = \langle\rangle \end{cases} \\
\ell(\text{LEFT}(q)) &= j_{st} \\
\ell(\text{RIGHT}(q)) &= k_{st}
\end{aligned}$$

The time complexity for this constrained version of the algorithm drops from $\Theta(N^3T^3V^3)$ to $\Theta(TV^3)$.

Partial Parse Trees A more realistic in-between scenario occurs when partial parse information is available for one or both of the languages. Special cases of particular interest include applications where bracketing or word alignment constraints may be derived from external sources beforehand. For

[3]Direct comparison with *word_align* should be avoided, however, since it is intended to work on corpora whose sentences are not aligned.

m	1	2	3	4	5	6	7	8	9	10	11	12
%	100	100	100	99.04	94.83	86.07	73.35	58.51	43.70	30.62	20.18	12.55

Figure 5: Proportion of alignment configurations generable by ITGs between length-m sequences.

example, a broad-coverage English bracketer may be available. If such constraints are reliable, it would be wasteful to ignore them.

A straightforward extension to the original algorithm inhibits hypotheses that are inconsistent with given constraints. Any entries in the DP table corresponding to illegal sub-hypotheses—i.e., those that would violate the given bracket-nesting or word-alignment conditions—are pre-assigned negative infinity values during initialization indicating impossibility. During the DP phase, computation of these entries is skipped. Since their probabilities remain impossible throughout, the illegal sub-hypotheses will never participate in any ML bi-bracketing. The running time reduction in this case depends heavily on the domain. We have found this strategy to be useful for incorporating punctuation constraints.

9 Conclusion

The twin concepts of bilingual language modeling and bilingual parsing have been proposed. We have introduced a new formalism, the inversion transduction grammar, and surveyed a variety of its applications to extracting linguistic information from parallel corpora. Its amenability to stochastic formulation, useful flexibility with leaky and minimal grammars, and tractability for practical applications are desirable properties. Various tasks such as segmentation, word alignment and bracket annotation are naturally incorporated as subproblems, and a high degree of compatibility with conventional monolingual methods is retained. In conjunction with automatic procedures for learning word translation lexicons, SITGs bring relatively underexploited bilingual constraints to bear on the task of extracting linguistic information for languages less well-studied than English.

A current direction is to investigate automatic training of SITGs. We derive in Wu (1995c) an EM-based re-estimation method for SITGs, and describe preliminary experiments.

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-speechtagging 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.

EARLEY, JAY. 1970. An efficient context-free parsing algorithm. *Communications of the Association for Computing Machinery*, 13(2):94–102.

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.

GAZDAR, GERALD & CHRISTOPHER S. MELLISH. 1989. *Natural language processing in LISP: An introduction to computational linguistics*. Reading, MA: Addison-Wesley.

LEWIS, P. M. & R. E. STEARNS. 1968. Syntax-directed transduction. *Journal of the Association for Computing Machinery*, 15:465–488.

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.

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.

SAVITCH, WALTER J. 1982. *Abstract machines and grammars*. Boston: Little, Brown.

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. 1995a. An algorithm for simultaneously bracketing parallel texts by aligning words. In *Proceedings of the 33rd Annual Conference of the Association for Computational Linguistics*, Cambridge, Massachusetts. To appear.

WU, DEKAI. 1995b. Grammarless extraction of phrasal translation examples from parallel texts. In *Proceedings of the Sixth International Conference on Theoretical and Methodological Issues in Machine Translation*, Leuven, Belgium. To appear.

WU, DEKAI. 1995c. Trainable coarse bilingual grammars for parallel text bracketing. In *Proceedings of the Third Annual Workshop on Very Large Corpora*, Cambridge, Massachusetts. To appear.

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.