Chapter 7

Bracketing and Aligning Words and Constituents in Parallel Text using Stochastic Inversion Transduction Grammars

Dekai Wu
HKUST, Hong Kong

Keywords: parallel text alignment, bilingual language modeling, stochastic inversion transduction grammars, bilingual parsing

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 a variety of parallel corpus analysis applications. Aside from the bilingual orientation, three major features distinguish the formalism from the finite-state transducers more traditionally found in computational linguistics: 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. Analysis of the formalism's expressiveness suggests that it is particularly well-suited to model ordering shifts between languages, balancing needed flexibility against complexity constraints. We discuss a number of examples of how stochastic inversion transduction grammars bring bilingual constraints to bear upon problematic corpus analysis tasks such as segmentation, bracketing, phrasal alignment, and parsing.

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. Transduction grammar models, especially of the finite-state family, have long been known. However, the imposition of identical ordering constraints upon both streams severely restricts their applicability,

and thus transduction grammars have received relatively little attention in language modeling research. 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 transduction grammars.

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 (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. The formalism's uniform integration of various types of bracketing and alignment constraints is one of its chief strengths.

The paper is divided in two main parts. We begin in the first part below by laying out the basic formalism, then show that reduction to a normal form is possible. We then raise several desiderata on the expressiveness of any bilingual language modeling formalism in terms of its constituent matching flexibility, and discuss how the characteristics of the inversion transduction formalism are particularly suited to address these criteria. 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. In the second part, we survey a number of sample applications and extensions of bilingual parsing for segmentation, bracketing, phrasal alignment, and parsing tasks.

2. INVERSION TRANSDUCTION GRAMMARS

A *transduction grammar* describes a structurally correlated pair of languages. For our purposes, the generative view is most convenient: the grammar generates transductions, so that two output streams are simultaneously generated, one for each language. This contrasts with the common input-output view popularized by both syntax-directed transduction grammars and finite-state transducers. The generative view is more appropriate here because for our applications the two languages' role is symmetric, in contrast to the usual applications of syntax-

directed transduction grammars. Moreover, the input-output view works better when a machine for accepting one of the languages (the input language) has a high degree of determinism, which is not the case here.

Our transduction model is context-free, rather than finite-state. Finite-state transducers, or FSTs, are well-known to be useful for specific tasks such as analysis of inflectional morphology (Koskenniemi 1983), text-to-speech conversion (Kaplan & Kay 1994), and nominal, number, and temporal phrase normalization (Gazdar & Mellish 1989). FSTs may also be used to parse restricted classes of context-free grammars (Pereira 1991, Roche 1994, Laporte 1996). However, the bilingual corpus analysis tasks we consider in this paper are quite different from the tasks for which FSTs are apparently well-suited. Our domain is broader, and the model possesses very little *a priori* specific structural knowledge of the language.

As a stepping stone to inversion transduction grammars, we first consider what a context-free model known as a *simple transduction grammar* (Lewis & Stearns 1968) would look like. Simple transduction grammars (as well as inversion transduction grammars) are restricted cases of the general class of context-free *syntax-directed* transduction grammars (Aho & Ullman 1969a, Aho & Ullman 1969b, Aho & Ullman 1972); however, we tend to avoid the term "syntax-directed" here, so as to de-emphasize the input-output connotation as discussed above.

A simple transduction grammar can be written by marking every terminal symbol for a particular output stream. Thus, each rewrite rule emits not one but two streams. For example, a rewrite rule of the form $A \rightarrow B\ x_1\ y_2\ C\ z_1$ 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 simple transduction grammar to model the generation of bilingual sentence pairs. As a mnemonic convention, we usually use the alternative notation $A \rightarrow B\ x/y\ C\ z/\varepsilon$ to associate matching output tokens. Though this additional information has no formal generative effect, it reminds us that x/y must be a 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 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 simple 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. [[[[[财政司]$_{NN}$]$_{NP}$ 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, simple 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。
(Authority will to Financial Secretary accountable.)

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

(6) VP → <VV PP>

Figure 1. (a) A simple transduction grammar. (b) An inverted-orientation production.

To make transduction grammars truly useful for bilingual tasks, we must escape the rigid parallel ordering constraint of simple 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.

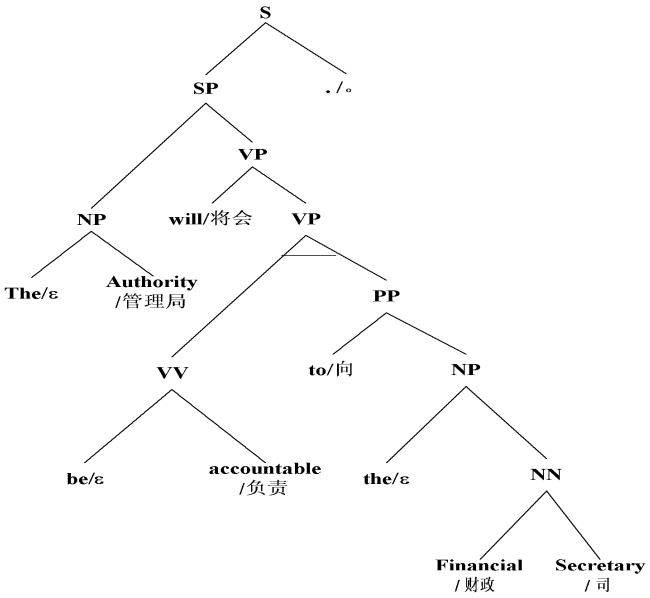

Figure 2. Inversion transduction grammar parse tree.

The inversion transduction grammar (ITG) formalism only minimally extends the generative power of a simple transduction grammar,[1] yet turns out to be surprisingly effective. Like simple transduction grammars, ITGs remain a subset of context-free (syntax-directed) transduction grammars (Lewis & Stearns 1968) but this view is too general to be of much help.[2] The productions of an inversion transduction grammar are interpreted just as in a simple transduction grammar, except that two possible *orientations* are allowed. Pure simple 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 $[A\ B]$ yields the string-pair $[C_1\ C_2]$ where $C_1 = A_1\ B_1$ and $C_2 = A_2\ B_2$. But the operator ⟨ ⟩ concatenates constituents on output stream

[1] The expressiveness of simple transduction grammars is equivalent to nondeterministic pushdown transducers (Savitch 1982).

[2] Also keep in mind that ITGs turn out to be especially suited for bilingual parsing applications, whereas pushdown transducers and syntax-directed transduction grammars are designed for monolingual parsing (in tandem with generation).

1 while reversing them on stream 2, so that $C_1 = A_1\ B_1$ but $C_2 = B_2\ A_2$. 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 simple transduction grammar, sentence-pair (4-5) 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 ⟨ ⟩ notation:

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

Alternatively, a graphical parse tree notation is shown in Figure 2, where the ⟨ ⟩ 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 labeled 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 by-product.

The nonterminals may not always look like those of an ordinary CFG. 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. One might even decide to choose nonterminals for an ITG that do not match linguistic categories, sacrificing this to the goal of ensuring that all corresponding substrings can be aligned.

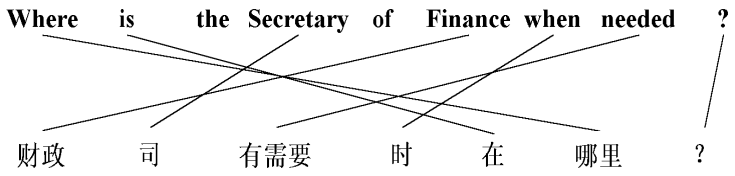

Figure 3. An extremely distorted alignment that can still be accommodated by an ITG.

An ITG can accommodate a wider range of ordering variation between the languages than might appear at first blush, through appropriate decomposition

of productions (and thus constituents), in conjunction with introduction of new auxiliary nonterminals where needed. For instance, even 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〉]〉 ?/?]

This bracketing is of course linguistically implausible, so whether such parses are acceptable depends on one's objective. Moreover, it may even remain possible to align constituents for phenomena whose underlying structure is not context-free – say, ellipsis or coordination – so long as the surface structures of the two languages fortuitously parallel each other (though again the bracketing would be linguistically implausible). We will return to the subject of ITGs' ordering flexibility in Section 4.

We stress again that the primary purpose of ITGs is to maximize robustness for parallel corpus analysis rather than to verify grammaticality, 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 1995). As the applications below demonstrate, the bilingual lexical constraints carry greater importance than the tightness of the grammar.

Formally, an inversion transduction grammar, or ITG, is denoted by $G = \left(\overline{N}, \overline{W}_1, \overline{W}_2, \overline{R}, \overline{S}\right)$, where $\overline{N}$ is a finite set of nonterminals, $\overline{W}_1$ is a finite set of words (terminals) of language 1, $\overline{W}_2$ is a finite set of words (terminals) of language 2, $\overline{R}$ is a finite set of rewrite rules (productions), and S is the start symbol. The space of word-pairs (terminal-pairs) $\bar{\mathrm{X}} = (\bar{W}_1 \cup \{\in\}) \times (\bar{W}_2 \cup \{\in\})$ contains lexical translations denoted *x*/*y* and singletons denoted x/ε or ε/y, where $x \in \overline{W}_1$ and $y \in \overline{W}_2$. Each production is either of straight orientation written $A \rightarrow [a_1 a_2 \ldots a_r]$, or of inverted orientation written $A \rightarrow \langle a_1 a_2 \ldots a_r \rangle$, where $a_i \in \overline{N} \cup \overline{X}$ and *r* is the rank of the production. The set of transductions generated by *G* is denoted *T*(*G*). The sets of (monolingual) strings generated by *G* for the first and second output languages are denoted $L_1(G)$ and $L_2(G)$, respectively.

3. A NORMAL FORM FOR INVERSION TRANSDUCTION GRAMMARS

We now show that every ITG can be expressed as an equivalent ITG in a 2-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 availability of a 2-normal form is a noteworthy characteristic of ITGs; no such normal form is available for unrestricted context-free (syntax-directed) transduction grammars (Aho & Ullman 1969b). The proof closely follows that for standard CFGs, and 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* $\varepsilon \in L_1(G)$ *and* $\varepsilon \in L_2(G)$, *then G' contains a single production of the form* $S' \rightarrow \varepsilon / \varepsilon$, *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* $A \rightarrow \varepsilon / \varepsilon$.

Lemma 2 *For any inversion transduction grammar G, there exists an equivalent inversion transduction grammar G' where 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 non-terminals.*

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:*

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

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

For each production of the form $A \rightarrow [B_1 \ldots B_n]$ we introduce new nonterminals $X_1 \ldots X_{n-2}$ in order to replace the production with the set of rules $A \rightarrow [B_1\ X_1]$, $X_1 \rightarrow [B_2\ X_2]$, … , $X_{n-3} \rightarrow [B_{n-2}\ X_{n-2}]$, $X_{n-2} \rightarrow [B_{n-1}\ B_n]$. Let $(\vec{e},\vec{c})$ be any string-pair derivable from $A \rightarrow [B_1 \ldots B_n]$, where $\vec{e}$ is output on stream 1 and $\vec{c}$ on stream 2. Define $\vec{e}^{\,i}$ as the substring of $\vec{e}$ derived from B_i, and similarly define $\vec{c}^{\,i}$. Then X_i generates $(\vec{e}^{\,i+1} \cdots \vec{e}^{\,n}, \vec{c}^{\,i+1} \cdots \vec{c}^{\,n})$ for all $1 \le i < n-1$, so the new production $A \rightarrow [B_1\ X_1]$ also generates $(\vec{e},\vec{c})$. 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 $A \rightarrow \langle B_1 \ldots B_n \rangle$ we replace the production with the set of rules $A \rightarrow \langle B_1\ Y_1 \rangle$, $Y_1 \rightarrow \langle B_2\ Y_2 \rangle$, … , $Y_{n-3} \rightarrow \langle B_{n-2}\ Y_{n-2} \rangle$, $Y_{n-2} \rightarrow \langle B_{n-1}\ B_n \rangle$. Let $(\vec{e},\vec{c})$ be any string-pair derivable from $A \rightarrow \langle B_1 \ldots B_n \rangle$, where $\vec{e}$ is output on stream 1 and $\vec{c}$ on stream 2. Again define $\vec{e}^{\,i}$ and $\vec{c}^{\,i}$ as the substrings derived from the B_i, but in this case $(\vec{e},\vec{c}) = (\vec{e}^{\,1} \cdots \vec{e}^{\,n}, \vec{c}^{\,n} \cdots \vec{c}^{\,1})$. Then Y_i generates $(\vec{e}^{\,i+1} \cdots \vec{e}^{\,n}, \vec{c}^{\,n} \cdots \vec{c}^{\,i+1})$ for all $1 \le i < n-1$, so the new production $A \rightarrow \langle B_1\ Y_1 \rangle$ also generates $(\vec{e},\vec{c})$. Again no additional string-pairs are generated due to the new productions. □

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

4. EXPRESSIVENESS CHARACTERISTICS

We now turn to the expressiveness desiderata for a matching formalism. It is of course difficult to make precise claims as to what characteristics are necessary and/or sufficient for such a model, since no cognitive studies that are directly pertinent to bilingual constituent alignment are available. Nonetheless, most related previous parallel corpus analysis models share certain conceptual approaches with ours, loosely based on cross-linguistic theories related to constituency, case frames, or thematic roles, as well as computational feasibility needs. Below we survey the most common constraints and discuss their relation to ITGs.

4.1 Crossing Constraints

Crossing constraints prohibit arrangements where the matchings between subtrees cross each another, unless the subtrees' immediate parent constituents are also matched to each other. For example, given the constituent matchings depicted as solid lines in Figure 4, the dotted-line matchings corresponding to potential lexical translations would be ruled illegal. Crossing constraints are implicit in many phrasal matching approaches, both constituency-oriented (Kaji, Kida, & Morimoto 1992; Cranias, Papageorgiou, & Piperidis 1994; Grishman 1994) and dependency-oriented (Sadler & Vendelmans 1990; Matsumoto, Ishimoto, & Utsuro 1993). The theoretical cross-linguistic hypothesis here is that the core arguments of frames tend to stay together over different languages. The constraint is also useful for computational reasons, since it helps avoid exponential bilingual matching times.

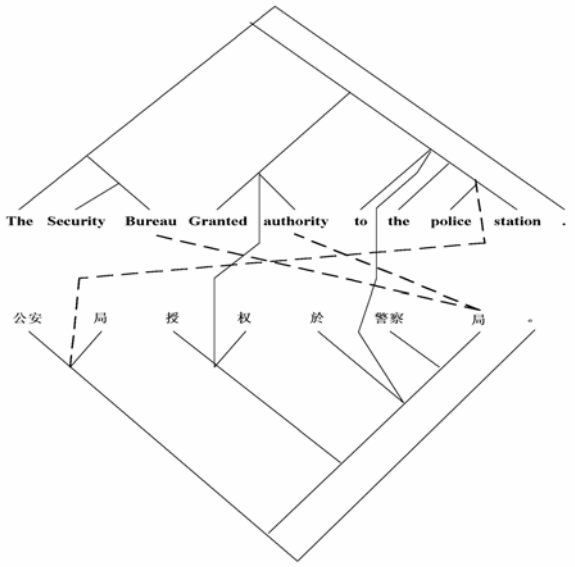

Figure 4. The crossing constraint (see text).

ITGs inherently implement a crossing constraint. The version of the crossing constraint as enforced by ITGs is actually even stronger. This is because even within a single constituent, the immediate subtrees are only permitted to cross in exact inverted order. As we shall argue below, this restriction reduces matching flexibility in a desirable fashion.

4.2 Rank Constraints

The second expressiveness desideratum for a matching formalism is to somehow limit the rank of constituents (the number of children or right-hand side

symbols), which dictates the span over which matchings may cross. As the number of subtrees of a L_1-constituent grows, the number of possible matchings to subtrees of the corresponding L_2-constituent grows combinatorially, with corresponding time complexity growth on the matching process. Moreover, if constituents can immediately dominate too many tokens of the sentences, the crossing constraint loses effectiveness – in the extreme, if a single constituent immediately dominates the entire sentence-pair, then any permutation is permissible without violating the crossing constraint. Thus we would like to constrain the rank as much as possible, while still permitting some reasonable degree of permutation flexibility.

Recasting this issue in terms of the general class of context-free (syntax-directed) transduction grammars, the number of possible subtree matchings for a single constituent grows combinatorially with the number of symbols on a production's right-hand side. However, it turns out that the ITG restriction of allowing only matchings with straight or inverted orientation effectively cuts the combinatorial growth, while still maintaining flexibility where needed.

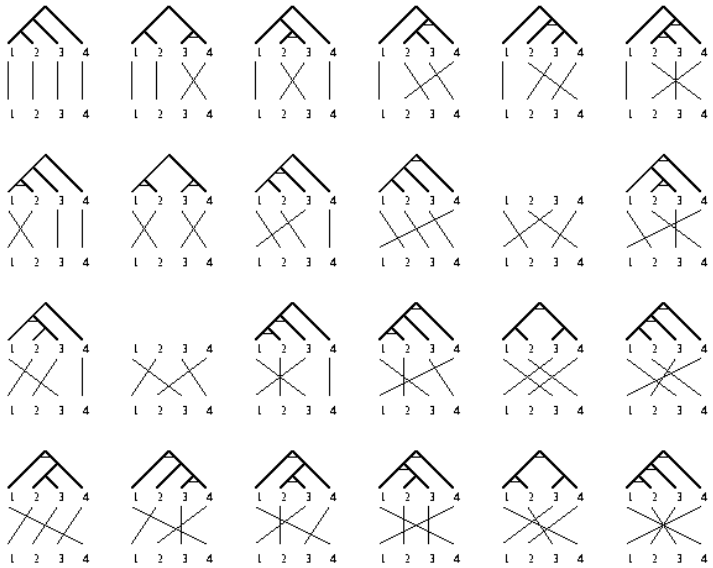

Figure 5. The 24 complete matchings of length four, with ITG parses for 22.

To see how ITGs maintain needed flexibility, consider Figure 5, which shows all 24 possible complete matchings between two constituents of length four each. Nearly all of these – 22 out of 24 – can be generated by an ITG as

shown by the parse trees (whose nonterminal labels are omitted).[3] The 22 permitted matchings are representative of real word-order transpositions between the English-Chinese sentences in our data. The only two matchings that cannot be generated are very distorted transpositions that we might call "inside-out" matchings. We have been unable to find real examples in our data of constituent arguments undergoing "inside-out" transposition.

Note that this hypothesis is for fixed word-order languages that are lightly inflected, such as English and Chinese. It would not be expected to hold for so-called scrambling or free word-order languages, or heavily inflected languages. However, inflections provide alternative surface cues for determining constituent roles (and thereby matchings), so it would not be necessary to apply ITG model to such languages. On the other hand, to see how ITGs cut combinatorial growth, consider the table in Figure 6, which compares growth in the number of legal complete matchings on a pair of subconstituent sequences. The third column shows the number of all possible complete matchings between two constituents with a rank of *r* subconstituents each (therefore this is also the behavior for unconstrained context-free (syntax-directed) transduction grammars). Compare this against the second column, which shows the number of complete matchings that can be accepted by an ITG between a pair of length-*r* sequences of subconstituents. The fourth column shows the proportion of matchings that ITGs can accept. Flexibility is nearly total for sequences of up to $r \leq 4$ subconstituents, with a rapid drop thereafter corresponding to the elimination of undesirably tangled (i.e., non-compositional) matchings.

Figure 7 shows the same numbers over all possible matchings, both complete and partial; in other words, for the more realistic case where some subconstituents are permitted to remain unmatched as singletons. The same desirable behavior is exhibited. The expressiveness of ITGs thus appears inherently suited to the degree of flexibility versus constraints needed for constituent matching.

[3] As discussed later, in many cases more than one parse tree can generate the same subconstituent matching. The trees shown are the canonical parses, as generated by the grammar of Figure 10.

R	ITG	ALL MATCHINGS	RATIO
0	1	1	1.000
1	1	1	1.000
2	2	2	1.000
3	6	6	1.000
4	22	24	0.917
5	90	120	0.750
6	394	720	0.547
7	1806	5040	0.358
8	8558	40320	0.212
9	41586	362880	0.115
10	206098	3628800	0.057
11	1037718	39916800	0.026
12	5293446	479001600	0.011
13	27297738	6227020800	0.004
14	142078746	87178291200	0.002
15	745387038	1307674368000	0.001
16	3937603038	20922789888000	0.000

Figure 6. Growth in number of legal *complete* subconstituent matchings for context-free (syntax-directed) transduction grammars with rank *r*, versus ITGs on a pair of subconstituent sequences of length *r* each.

R	ITG	ALL MATCHINGS	RATIO
0	1	1	1.000
1	2	2	1.000
2	7	7	1.000
3	34	34	1.000
4	207	209	0.990
5	1466	1546	0.948
6	11471	13327	0.861
7	96034	130922	0.734
8	843527	1441729	0.585
9	7678546	17572114	0.437
10	71852559	234662231	0.306
11	687310349	3405357682	0.202
12	6693544171	53334454417	0.126
13	66167433658	896324308634	0.074
14	662393189919	16083557845279	0.041
15	6703261197506	306827170866106	0.022
16	68474445473303	6199668952527617	0.011

Figure 7. Growth in number of all legal subconstituent matchings (complete or partial, meaning that some subconstituents are permitted to remain unmatched as singletons) for context-free (syntax-directed) transduction grammars with rank *r*, versus ITGs on a pair of subconstituent sequences of length *r* each.

5. STOCHASTIC INVERSION TRANSDUCTION GRAMMARS

In a stochastic ITG (SITG), a probability is associated with each rewrite rule. Following the standard convention, we use *a* and *b* to denote probabilities for syntactic and lexical rules, respectively. For example, the probability of the rule $NN \xrightarrow{0.4} [A\ N]$ is $a_{NN \to [A\ N]} = 0.4$. The probability of a lexical rule $A \xrightarrow{0.001} x/y$ is $b_A(x,y) = 0.001$. Let W_1, W_2 be the vocabulary sizes of the two languages, and $\overline{N} = \{A_1, \ldots, A_N\}$ be the set of nonterminals with indices $1, \ldots, \overline{N}$. (For conciseness, we sometimes abuse the notation by writing an index when we mean the corresponding nonterminal symbol, as long as this introduces no confusion.) Then for every $1 \le i \le N$, the production probabilities are subject to the constraint that

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

We now introduce an algorithm for parsing with stochastic ITGs, that computes an optimal parse given a sentence-pair using dynamic programming. 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 CYK parsing (Kasami 1965; Younger 1967).

Let the input English sentence be $\vec{e}_1, \ldots, \vec{e}_T$ and the corresponding input Chinese sentence be $\vec{c}_1, \ldots, \vec{c}_V$. As an abbreviation we write $\vec{e}_{s..t}$ for the sequence of words $\vec{e}_{s+1}, \vec{e}_{s+2}, \ldots, \vec{e}_t$, and similarly for $\vec{c}_{u..v}$; also, $\vec{e}_{s..s} = \varepsilon$ is the empty string. 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 $\vec{e}_{s..t}$ and $\vec{c}_{u..v}$ 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

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

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

The algorithm computes $\delta_{0,T,0,V}(S)$ using the following recurrences. Note that we generalize 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

$$\delta_{t-1,t,v-1,v}(i) = b_i(\vec{e}_t/\vec{c}_v), \quad \begin{matrix}1\le t\le T\\ 1\le v\le V\end{matrix} \tag{1}$$

$$\delta_{t-1,t,v,v}(i) = b_i(\vec{e}_t/\varepsilon), \quad \begin{matrix}1\le t\le T\\ 0\le v\le V\end{matrix} \tag{2}$$

$$\delta_{t,t,v-1,v}(i) = b_i(\varepsilon/\vec{c}_v), \quad \begin{matrix}0\le t\le T\\ 1\le v\le V\end{matrix} \tag{3}$$

2. RECURSION

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

$$\delta_{stuv}(i) = \max[\delta^{[\,]}_{stuv}(i), \delta^{\langle\rangle}_{stuv}(i)] \tag{4}$$

$$\theta_{stuv}(i) = \begin{cases}[\,] & \text{if } \delta^{[\,]}_{stuv}(i) > \delta^{\langle\rangle}_{stuv}(i)\\ \langle\rangle & \text{otherwise}\end{cases} \tag{5}$$

where

$$\delta^{[\,]}_{stuv}(i) = \max_{\substack{1\le j\le N\\ 1\le k\le N\\ s\le S\le t\\ u\le U\le v\\ (S-s)(t-S)+(U-u)(v-U)\neq 0}} a_{i\to[jk]}\delta_{sSuU}(j)\delta_{StUv}(k) \tag{6}$$

$$\begin{bmatrix} \iota^{[\,]}_{stuv}(i) \\ \kappa^{[\,]}_{stuv}(i) \\ \sigma^{[\,]}_{stuv}(i) \\ \nu^{[\,]}_{stuv}(i) \end{bmatrix} = \underset{\substack{1\le j\le N \\ 1\le k\le N \\ s\le S\le t \\ u\le U\le v \\ (S-s)(t-S)+(U-u)(v-U)\neq 0}}{\arg\max} a_{i\to[jk]}\delta_{sSuU}(j)\delta_{StUv}(k) \quad (7)$$

$$\delta^{\langle\rangle}_{stuv}(i) = \underset{\substack{1\le j\le N \\ 1\le k\le N \\ s\le S\le t \\ u\le U\le v \\ (S-s)(t-S)+(U-u)(v-U)\neq 0}}{\max} a_{i\to\langle jk\rangle}\delta_{sSUv}(j)\delta_{StuU}(k) \quad (8)$$

$$\begin{bmatrix} \iota^{\langle\rangle}_{stuv}(i) \\ \kappa^{\langle\rangle}_{stuv}(i) \\ \sigma^{\langle\rangle}_{stuv}(i) \\ \nu^{\langle\rangle}_{stuv}(i) \end{bmatrix} = \underset{\substack{1\le j\le N \\ 1\le k\le N \\ s\le S\le t \\ u\le U\le v \\ (S-s)(t-S)+(U-u)(v-U)\neq 0}}{\arg\max} a_{i\to\langle jk\rangle}\delta_{sSUv}(j)\delta_{StuU}(k) \quad (9)$$

3. RECONSTRUCTION

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

$$LEFT(q) = \begin{cases} NIL & \text{if } t-s+v-u \le 2 \\ (s,\sigma^{[\,]}_q(l(q)),u,\nu^{[\,]}_q(l(q))) & \text{if } \theta_q(l(q)) = [\,] \text{ and } t-s+v-u > 2 \\ (s,\sigma^{\langle\rangle}_q(l(q)),\nu^{\langle\rangle}_q(l(q)),v) & \text{if } \theta_q(l(q)) = \langle\rangle \text{ and } t-s+v-u > 2 \end{cases} \quad (10)$$

$$RIGHT(q) = \begin{cases} NIL & \text{if } t-s+v-u \le 2 \\ (\sigma^{[\,]}_q(l(q)),t,\nu^{[\,]}_q(l(q)),v) & \text{if } \theta_q(l(q)) = [\,] \text{ and } t-s+v-u > 2 \\ (\sigma^{\langle\rangle}_q(l(q)),t,u,\nu^{\langle\rangle}_q(l(q))) & \text{if } \theta_q(l(q)) = \langle\rangle \text{ and } t-s+v-u > 2 \end{cases} \quad (11)$$

$$l(LEFT(q)) = \iota^{\theta_q(l(q))}_q(l(q)) \quad (12)$$

$$l(RIGHT(q)) = \kappa^{\theta_q(l(q))}_q(l(q)) \quad (13)$$

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.

6. 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 (5) demonstrate why the obvious trick of taking single characters as words is not a workable strategy.) The usual Chinese NLP architecture first pre-processes input text through a word segmentation module (Chiang *et al.* 1992; Lin, Chiang, & Su 1992; Chang & Chen 1993; Lin, Chiang, & Su 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

$$\delta^0_{stuv}(i) = b_i(\vec{e}_{s..t}/\vec{c}_{u..v}), \quad \begin{matrix} 0 \le s \le t \le T \\ 0 \le u \le v \le V \end{matrix} \tag{14}$$

2. RECURSION

$$\delta_{stuv}(i) = \max[\delta^{[\,]}_{stuv}(i), \delta^{\langle\rangle}_{stuv}(i), \delta^{0}_{stuv}(i)] \tag{15}$$

$$\theta_{stuv}(i) = \begin{cases} [\,] & \text{if } \delta^{[\,]}_{stuv}(i) > \delta^{\langle\rangle}_{stuv}(i) \text{ and } \delta^{[\,]}_{stuv}(i) > \delta^{0}_{stuv}(i) \\ \langle\rangle & \text{if } \delta^{\langle\rangle}_{stuv}(i) > \delta^{[\,]}_{stuv}(i) \text{ and} \delta^{\langle\rangle}_{stuv}(i) > \delta^{0}_{stuv}(i) \\ 0 & \text{otherwise} \end{cases} \tag{16}$$

3. RECONSTRUCTION

$$LEFT(q) = \begin{cases} NIL & \text{if } t - s + v - u \le 2 \\ (s, \sigma^{[\,]}_q(l(q)), u, v^{[\,]}_q(l(q))) & \text{if } \theta_q(l(q)) = [\,] \text{ and } t - s + v - u > 2 \\ (s, \sigma^{\langle\rangle}_q(l(q)), v^{\langle\rangle}_q(l(q)), v) & \text{if } \theta_q(l(q)) = \langle\rangle \text{ and } t - s + v - u > 2 \\ NIL & \text{otherwise} \end{cases} \tag{17}$$

$$RIGHT(q) = \begin{cases} NIL & \text{if } t - s + v - u \le 2 \\ (\sigma^{[\,]}_q(l(q)), t, v^{[\,]}_q(l(q)), v) & \text{if } \theta_q(l(q)) = [\,] \text{ and } t - s + v - u > 2 \\ (\sigma^{\langle\rangle}_q(l(q)), t, u, v^{\langle\rangle}_q(l(q))) & \text{if } \theta_q(l(q)) = \langle\rangle \text{ and } t - s + v - u > 2 \\ NIL & \text{otherwise} \end{cases} \tag{18}$$

In our experience this method has proven extremely effective for avoiding mis-segmentation 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.

7. 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, Garside, & Leech 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 gram-

mar is available means that constituent categories are not differentiated. Instead, a generic *bracketing 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:

$A \xrightarrow{a} [A\ A]$
$A \xrightarrow{a} \langle A\ A \rangle$
$A \xrightarrow{b_{ij}} u_i / v_j$ for all *i,j* English-Chinese lexical translations
$A \xrightarrow{b_{i\varepsilon}} u_i / \varepsilon$ for all *i* English vocabulary
$A \xrightarrow{b_{\varepsilon j}} \varepsilon / v_i$ for all *j* Chinese vocabulary

Longer productions with rank > 2 are not needed; we show in the subsections below that this minimal transduction grammar in normal form is generatively equivalent to any reasonable bracketing transduction grammar. Moreover, we also show how postprocessing using rotation and flattening operations restores the rank flexibility so that an output bracketing can hold more than two immediate constituents, as shown in Figure 11.

The b_{ij} distribution actually encodes the English-Chinese translation lexicon with degrees of probability on each potential word translation. 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 $b_{i\varepsilon}$ and $b_{\varepsilon j}$, so that the optimal bracketing resorts to these productions only when it is otherwise impossible to match the singletons. The parameter *a* here is of no practical effect, and is chosen to be very small relative to the b_{ij} probabilities of lexical translation pairs. The result is that the maximum-likelihood parser selects the parse tree that best meets the combined lexical translation preferences, as expressed by the b_{ij} probabilities.

Several additional methods are useful for improving accuracy by incorporating pre/post-positional biases and flattening the bracketings in cases where there is no cross-lingual discrimination to increase the certainy between alternative bracketings. Space does not permit description here; see Wu (1997).

Using these methods, 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 ab-

sent 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 11.

[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/ 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/我們 ε/的 prosperity/繁榮] . /。]

Figure 11. Bracketing output examples. (<> = unrecognized input token.)

The bracket precision was 80% for the English sentences, and 78% 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 9.

8. PHRASAL ALIGNMENT

8.1 Phrasal Alignment

Phrasal translation examples at the subsentential level are an essential resource for many MT and machine-assisted translation architectures. This requirement is becoming increasingly direct for the example-based machine translation paradigm (Nagao 1984), whose translation flexibility is strongly restricted if the examples are only at the sentential level. It can now be assumed that a parallel bilingual corpus may be aligned to the sentence level with reasonable accuracy

(Kay & Röscheisen 1988; Catizone, Russell, & Warwick 1989; Gale & Church 1991; Brown, Lai, & Mercer 1991; Chen 1993), even for languages as disparate as Chinese and English (Wu 1994). Algorithms for subsentential alignment have been developed as well at granularities of the character (Church 1993), word (Dagan, Church, & Gale 1993; Fung & Church 1994; Fung & McKeown 1994), collocation (Smadja 1992), and specially-segmented (Kupiec 1993) levels. However, the identification of subsentential, nested, phrasal translations within the parallel texts remains a non-trivial problem, due to the added complexity of dealing with constituent structure. Manual phrasal matching is feasible only for small corpora, either for toy-prototype testing or for narrowly-restricted applications.

Automatic approaches to identification of subsentential translation units have largely followed what we might call a ``parse-parse-match'' procedure. Each half of the parallel corpus is first parsed individually using a monolingual grammar. Subsequently, the constituents of each sentence-pair are matched according to some heuristic procedure. A number of recent proposals can be cast in this framework (Sadler & Vendelmans 1990; Kaji, Kida, & Morimoto 1992; Matsumoto, Ishimoto, & Utsuro 1993; Cranias, Papageorgiou, & Piperidis 1994; Grishman 1994).

The "parse-parse-match" procedure is susceptible to three weaknesses:

– *Appropriate, robust, monolingual grammars may not be available.* This condition is particularly relevant for many non-Western-European languages such as Chinese. A grammar for this purpose must be robust since it must still identify constituents for the subsequent matching process even for unanticipated or ill-formed input sentences.

– *The grammars may be incompatible across languages.* The best-matching constituent types between the two languages may not include the same core arguments. While grammatical differences can make this problem unavoidable, there is often a degree of arbitrariness in a grammar's chosen set of syntactic categories, particularly if the grammar is designed to be robust. The mismatch can be exacerbated when the monolingual grammars are designed independently, or under different theoretical considerations.
– *Selection between multiple possible arrangements may be arbitrary.* By an ``arrangement'' between any given pair of sentences from the parallel corpus, we mean a set of matchings between the constituents of the sentences. The problem is that in some cases, a constituent in one sentence may have several potential matches in the other, and the matching heuristic may be unable to discriminate between the options. In the sentence pair of Figure 4, for example, both *Security Bureau* and *police station* are potential lexical matches to ¤½¦w§½. To choose the best set of matchings, an optimization over some measure of overlap between the structural analysis of the two sentences is

needed. Previous approaches to phrasal matching employ arbitrary heuristic functions on, say, the number of matched subconstituents.

Our method attacks the weaknesses of the "parse-parse-match"procedure by using (1) only a translation lexicon with no language-specific grammar, (2) a bilingual rather than monolingual formalism, and (3) a probabilistic formulation for resolving the choice between candidate arrangements. The approach differs in its single-stage operation that simultaneously chooses the constituents of each sentence and the matchings between them.

The raw phrasal translations suggested by the parse output were then filtered to remove those pairs containing more than 50% singletons, since such pairs are likely to be poor translation examples. Examples that occurred more than once in the corpus were also filtered out, since repetitive sequences in our corpus tend to be non-grammatical markup. This yielded approximately 2,800 filtered phrasal translations, some examples of which are shown in Figure 12. A random sample of the phrasal translation pairs was then drawn, giving a precision estimate of 81.5%.

1% in real	1%的實質
Would you	你是否
an acceptable starting point for this new policy	是可接受爲這項新政策的起點
are about 3.5 million	大概有350萬
born in Hong	在香港出生
for Hong	爲香港
have the right to decide our	有權決定我
in what way the Government would increase their job opportunities; and	政府如何增加他們的就業機會；及
last month	上個月
never to say "never"	不要說"永不"
reserves and surpluses	儲備和盈餘
starting point for this new policy	爲這項新政策的起點
there will be many practical difficulties in terms of implementation	實行時會有很多實際困難
year ended 31 March 1991	截至一九九一年一月三十一日

Figure 12. Examples of extracted phrasal translations.

Although this already represents a useful level of accuracy, it does not in our opinion reflect the full potential of the formalism. Inspection revealed that performance was greatly hampered by our noisy translation lexicon which was automatically learned; it could be manually post-edited to reduce errors. Commercial online translation lexicons could also be employed if available. Higher precision could be also achieved without great effort by engineering a small number of broad nonterminal categories. This would reduce errors for known idiosyncratic patterns, at the cost of manual rule building.

The automatically extracted phrasal translation examples are especially useful where the phrases in the two languages are not compositionally derivable solely from obvious word transltions. An example is [have acquired/ε ε/學到 new/新 skills/技能] in Figure 11. The same 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.

8.2 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, Church, & Gale 1993; Dagan & Church 1994), a penalty is imposed according to the deviation from an ideal matching, as constructed by linear interpolation.

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 7 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 length-4 sequences and a rapid drop-off thereafter. 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.

9. BILINGUAL CONSTRAINT TRANSFER

9.1 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 $e_{0..T}$ into two constituent subtrees deriving $e_{0..S}$ and $e_{S..T}$ 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 $e_{s..t}$. The following simplifications can then be made to the parsing algorithm:

2. RECURSION

For all English constituents $\bar{e}_{s..t}$ and all i,u,v such that $\begin{cases} 1 \le i \le N \\ 0 \le u < v \le V \end{cases}$

$$\delta^{[\,]}_{stuv}(i) = \max_{u \le U \le v} a_{i \to [j_{st} k_{st}]} \delta_{s,S_{st},u,U}(j_{st}) \delta_{S_{st},t,U,v}(k_{st}) \tag{19}$$

$$v^{[\,]}_{stuv}(i) = \arg\max_{u \le U \le v} \delta_{s,S_{st},u,U}(j_{st}) \delta_{S_{st},t,U,v}(k_{st}) \tag{20}$$

$$\delta^{\langle\rangle}_{stuv}(i) = \max_{u \le U \le v} a_{i \to \langle j_{st} k_{st} \rangle} \delta_{s,S_{st},U,v}(j_{st}) \delta_{S_{st},t,u,U}(k_{st}) \tag{21}$$

$$v^{\langle\rangle}_{stuv}(i) = \arg\max_{u \le U \le v} \delta_{s,S_{st},U,v}(j_{st})\delta_{S_{st},t,u,U}(k_{st}) \tag{22}$$

3. RECONSTRUCTION

$$LEFT(q) = \begin{cases} (s, S_{st}, u, v^{[]}_q(l(q))) & \text{if } \vartheta_q(l(q)) = [\,] \\ (s, S_{st}, v^{\langle\rangle}_q(l(q)), v) & \text{if } \vartheta_q(l(q)) = \langle\rangle \end{cases} \tag{23}$$

$$RIGHT(q) = \begin{cases} (S_{st}, t, v^{[]}_q(l(q)), v) & \text{if } \vartheta_q(l(q)) = [\,] \\ (S_{st}, t, u, v^{\langle\rangle}_q(l(q))) & \text{if } \vartheta_q(l(q)) = \langle\rangle \end{cases} \tag{24}$$

$$l(LEFT(q)) = j_{st} \tag{25}$$

$$l(RIGHT(q)) = k_{st} \tag{26}$$

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

9.2 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 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 dynamic programming 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 recursion 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 constraints.

We have found this strategy to be useful for incorporating punctuation constraints. Certain punctuation characters give constituency indications with high reliability; "perfect separators" include colons and Chinese full stops, while "perfect delimiters" include parentheses and quotation marks.

10. UNRESTRICTED-FORM GRAMMARS

It is possible to construct a parser that accepts unrestricted-form, rather than normal form, grammars. In this case an Earley-style scheme (Earley 1970), employing an active chart, can be used. The time complexity remains the same as the normal-form case.

We have found this to be useful in practice. For bracketing grammars of the type considered in this paper, there is no advantage. However, for more complex, linguistically-structured grammars, the more flexible parser does not require the unreasonable numbers of productions that can easily arise from normal form requirements. For most grammars, we have found performance to be comparable or faster than the normal-form parser.

11. 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 correlations to bear on the task of extracting linguistic information for languages less well-studied than English.

We are currently pursuing several directions. We are developing an iterative training method based on expectation-maximization for estimating the probabilities from parallel training corpora. Also, in contrast to the applications discussed here, which deal with analysis and annotation of parallel corpora, we are working on incorporating the SITG model directly into our runtime translation architecture. The initial results indicate excellent performance gains.

12. ACKNOWLEDGEMENT

I would like to thank Xuanyin Xia, Eva Wai-Man Fong, Pascale Fung, and Derick Wood, as well as various anonymous reviewers.

13. REFERENCES

Aho, A. V. and J. D. Ullman. 1969a. Properties of syntax directed translations. *J. Computer and System Sciences*, 3(3):319-334.

Aho, A. V. and J. D. Ullman. 1969b. Syntax directed translations and the pushdown assembler. *J. Computer and System Sciences*, 3(1):37-56.

Aho, A. V. and J. D. Ullman. 1972. *The Theory of Parsing, Translation, and Compiling*. Prentice Hall, Englewood Cliffs, NJ.

Black, E., R. Garside, and G. Leech, editors. 1993. *Statistically-Driven Computer Grammars of English: The IBM/Lancaster Approach*. Editions Rodopi, Amsterdam.

Brown, P. F., J. Cocke, S. A. DellaPietra, V. J. DellaPietra, F. Jelinek, J. D. Lafferty, R. L. Mercer, and P. S. Roossin. 1990. A statistical approach to machine translation. *Computational Linguistics*, 16(2):29-85.

Brown, P. F., S. A. DellaPietra, V. J. DellaPietra, and R. L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263-311.

Brown, P. F., Jennifer C. Lai, and R. L. Mercer. 1991. Aligning sentences in parallel corpora. In *Proceedings of the 29th Annual Conference of the Association for Computational Linguistics*, pages 169-176, Berkeley.

Catizone, R., G. Russell, and S. Warwick. 1989. Deriving translation data from bilingual texts. In *Proceedings of the First International Acquisition Workshop*, Detroit.

Chang, Chao-Huang and Cheng-Der Chen. 1993. HMM-based part-of-speech tagging for Chinese corpora. In *Proceedings of the Workshop on Very Large Corpora*, pages 40-47, Columbus, Ohio, June.

Chen, Stanley F. 1993. Aligning sentences in bilingual corpora using lexical information. In *Proceedings of the 31st Annual Conference of the Association for Computational Linguistics*, pages 9-16, Columbus, OH.

Chiang, Tung-Hui, Jing-Shin Chang, Ming-Yu Lin, and Keh-Yih Su. 1992. Statistical models for word segmentation and unknown resolution. In *Proceedings of ROCLING-92*, pages 121-146.

Church, K. 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*, pages 1-8, Columbus, OH.

Cranias, L., H. Papageorgiou, and S. Piperidis. 1994. A matching technique in example-based machine translation. In *Proceedings of the Fifteenth International Conference on Computational Linguistics*, pages 100-104, Kyoto.

Dagan, I. and K. W. Church. 1994. Termight: Identifying and translating technical terminology. In *Proceedings of the Fourth Conference on Applied Natural Language Processing*, pages 34-40, Stuttgart, October.

Dagan, I., K. W. Church, and W. A. Gale. 1993. Robust bilingual word alignment for machine aided translation. In *Proceedings of the Workshop on Very Large Corpora*, pages 1-8, Columbus, OH, June.

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

Fung, Pascale and K. W. Church. 1994. K-vec: A new approach for aligning parallel texts. In *Proceedings of the Fifteenth International Conference on Computational Linguistics*, pages 1096-1102, Kyoto.

Fung, Pascale and K. 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*, pages 81-88, Columbia, Maryland, October.

Fung, Pascale and Dekai Wu. 1994. Statistical augmentation of a Chinese machine-readable dictionary. In *Proceedings of the Second Annual Workshop on Very Large Corpora*, pages 69-85, Kyoto, August.

Gale, W. A. and K. W. Church. 1991. A program for aligning sentences in bilingual corpora. In *Proceedings of the 29th Annual Conference of the Association for Computational Linguistics*, pages 177-184, Berkeley.

Gale, W. A., K. W. Church, and D. Yarowsky. 1992. Using bilingual materials to develop word sense disambiguation methods. In *TMI-92, Proceedings of the Fourth International Conference on Theoretical and Methodological Issues in Machine Translation*, pages 101-112, Montreal.

Gazdar, G. and C. S. Mellish. 1989. *Natural Language Processing in LISP: An Introduction to Computational Linguistics*. Addison-Wesley, Reading, MA.

Grishman, R. 1994. Iterative alignment of syntactic structures for a bilingual corpus. In *Proceedings of the Second Annual Workshop on Very Large Corpora*, pages 57-68, Kyoto, August.

Kaji, H., Y. Kida, and Y. Morimoto. 1992. Learning translation templates from bilingual text. In *Proceedings of the Fourteenth International Conference on Computational Linguistics*, pages 672-678, Nantes.

Kaplan, R. M. and M. Kay. 1994. Regular models of phonological rule systems. *Computational Linguistics*, 20(3):331-378.

Kasami, T. 1965. An efficient recognition and syntax analysis algorithm for context-free languages. Technical Report AFCRL-65-758, Air Force Cambridge Research Laboratory, Bedford, MA.

Kay, M. and M. Röscheisen. 1988. Text-translation alignment. Technical Report P90-00143, Xerox Palo Alto Research Center.

Koskenniemi, K. 1983. Two-level morphology: A general computational model for word-form recognition and production. Technical Report 11, Department of General Linguistics, University of Helsinki.

Kupiec, J. 1993. An algorithm for finding noun phrase correspondences in bilingual corpora. In *Proceedings of the 31st Annual Conference of the Association for Computational Linguistics*, pages 17-22, Columbus, OH.

Laporte, E. 1996. Context-free parsing with finite-state transducers. In *String Processing Colloqium*, Recife, Brazil.

Lewis, P. M. and R. E. Stearns. 1968. Syntax-directed transduction. *Journal of the Association for Computing Machinery*, 15:465-488.

Lin, Ming-Yu, Tung-Hui Chiang, and Keh-Yih Su. 1993. A preliminary study on unknown word problem in Chinese word segmentation. In *Proceedings of ROCLING-93*, pages 119-141.

Lin, Yi-Chung, Tung-Hui Chiang, and Keh-Yih Su. 1992. Discrimination oriented probabilistic tagging. In *Proceedings of ROCLING-92*, pages 85-96.

Magerman, D. M. and M. P. Marcus. 1990. Parsing a natural language using mutual information statistics. In *Proceedings of AAAI-90, Eighth National Conference on Artificial Intelligence*, pages 984-989.

Matsumoto, Y., H. Ishimoto, and T. Utsuro. 1993. Structural matching of parallel texts. In *Proceedings of the 31st Annual Conference of the Association for Computational Linguistics*, pages 23-30, Columbus, OH.

Nagao, M. 1984. A framework of a mechanical translation between japanese and english by analogy principle. In Alick Elithorn and Ranan Banerji, editors, *Artificial and Human Intelligence: Edited Review Papers Presented at the International NATO Symposium on Artificial and Human Intelligence*. North-Holland, Amsterdam, pages 173-180.

Pereira, F. 1991. Finite-state approximation of phrase structure grammars. In *Proceedings of the 29th Annual Conference of the Association for Computational Linguistics*, Berkeley.

Pereira, F. and Y. Schabes. 1992. Inside-outside reestimation from partially bracketed corpora. In *Proceedings of the 30th Annual Conference of the Association for Computational Linguistics*, pages 128-135, Newark, DE.

Roche, E. 1994. Two parsing algorithms by means of finite-state transducers. In *Proceedings of the Fifteenth International Conference on Computational Linguistics*, Kyoto.

Sadler, V. and R. Vendelmans. 1990. Pilot implementation of a bilingual knowledge bank. In *Proceedings of the Thirteenth International Conference on Computational Linguistics*, pages 449-451, Helsinki.

Savitch, W. J. 1982. *Abstract Machines and Grammars*. Little, Brown, Boston.

Smadja, F. A. 1992. How to compile a bilingual collocational lexicon automatically. In *AAAI-92 Workshop on Statistically-Based NLP Techniques*, pages 65-71, San Jose, CA, July.

Sproat, R., Chilin Shih, W. A. Gale, and Nancy 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*, pages 66-72, Las Cruces, New Mexico, June.

Viterbi, A. 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*, pages 80-87, Las Cruces, New Mexico, June.

Wu, Dekai. 1995. An algorithm for simultaneously bracketing parallel texts by aligning words. In *Proceedings of the 33rd Annual Conference of the Association for Computational Linguistics*, pages 244-251, Cambridge, Massachusetts, June.

Wu, Dekai. 1997. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Computational Linguistics* **23**(3):377-404.

Wu, Dekai and Pascale Fung. 1994. Improving Chinese tokenization with linguistic filters on statistical lexical acquisition. In *Proceedings of the Fourth Conference on Applied Natural Language Processing*, pages 180-181, Stuttgart, October.

Wu, Dekai and Xuanyin Xia. 1994. Learning an English-Chinese lexicon from a parallel corpus. In *AMTA-94, Association for Machine Translation in the Americas*, pages 206-213, Columbia, Maryland, October.

Wu, Zimin and Gwyneth Tseng. 1993. Chinese text segmentation for text retrieval: Achievements and problems. *Journal of The American Society for Information Science*, 44(9):532-542.

Younger, D. H. 1967. Recognition and parsing of context-free languages in time . *Information and Control*, 10(2):189-208.