A Corpus-Based Statistics-Oriented Transfer and Generation Model for Machine Translation

†Jing-Shin Chang and ‡Keh-Yih Su

Department of Electrical Engineering
National Tsing-Hua University
Hsinchu, Taiwan 30043, R.O.C.
†shin@hera.ee.nthu.edu.tw and ‡kysu@bdc.com.tw

ABSTRACT

In this paper, a corpus-based approach for acquiring transfer rules and selecting the most preferred transfer between a language pair is proposed. A transfer score is defined to measure the preference of different mappings between the source-target sentence pair, and a generation score is defined to provide a probabilistic mechanism for finding the most preferred generation pattern. An algorithm is proposed to find the appropriate transfer units within a syntax tree and the corresponding transfer rules. By applying such an algorithm, the preferred generation rules can be learned directly from the target language so that the generation of the target sentences can be tuned to follow the grammar and style of the target language, instead of being bounded by the analysis grammar of the source language.

1. Overview of Transfer Models

In a transfer-based machine translation system, a source sentence is analyzed into an intermediate representation for the source language. The intermediate representation is then transferred into its target equivalent. Finally, the target surface strings are generated according to the intermediate representation of the target sentence. More specifically, the major tasks for transfer and generation include (1) reducing the analysis result into an intermediate form that is suitable for transfer, (2) selecting appropriate target words for source words, (3) making appropriate mapping from the source form to the target structure, and (4) generating the target equivalent from the target representation. (The word 'transfer' will sometimes be used ambiguously to refer to both transfer and generation.)

1.1. Rule-Based Approaches

A common transfer approach is to carry out a sequence of tree to tree mapping, either at syntactic level or semantic level, by using a set of source-target transfer pattern-action pairs to reflect the changes in substructures and linear order in the language pair [Benn 85, Naga 85, Tsut 90]. The major problems with such an approach are the coverage of the transfer rules (or patterns) and the consistency among the rules in the context of a wide variety of application domains. It is hard and costly to acquire a complete and consistent set of transfer rules manually. Notably, it is nontrivial to identify the appropriate *atomic units* for transfer. The large set of transfer rules thus imposes nontrivial acquisition and maintenance problems as the system scales up.

Additionally, in most transfer-based systems that follow a "one-way" analysis-transfer-generation process, the generated sentences are often strongly bounded to the *analysis* grammar since the generation rules are influenced greatly by the source language. The generation grammar might preserve much stylistic characteristics of the source language such that the generated sentences are unnatural to the native speakers [Su 93]. Our experiences with the BehaviorTran (formerly the ArchTran) MTS [Chen 91] show that such translation quality is still far below the user

expectation for publication even though they are considered 'correct' and 'understandable' [Su 90, 92a, 93]. One major reason for the quality gap is due to the fact that the generation grammar is not really designed from the target language's point of view. To reduce the human costs in transfer rule acquisition and to make the generation rules less source-bounded, a systematic approach for training the transfer rules from the source side and a method for acquiring the generation rules from the target side are desirable.

1.2. Example-Based Approaches

Example-based MT has been widely accepted as a potential approach for translation (and thus transfer and generation) in recent years [Nagao 84, Sato 90]. The translation process involves the matching of the input sentence against a large corpus of translation examples with certain "similarity" measure. The translation is then performed according to the translation example that is most similar to the input sentence. The general problem is: the "similarity" measure may not be able to be normalized to a uniform scale for comparison. Furthermore, a large example base may contain substantial amount of redundant information, which can be reduced by proper modelling. Therefore, it is desirable to reduce such redundancy and to reduce the searching time even if it is affordable to construct a large example base.

1.3. Purely Statistical Approaches

An alternative to the above transfer approaches is to adopt a translation model as proposed in [Brow 90]. The statistical approach, though theoretically interesting, has some technical difficulties. First, since [Brow 90] uses a highly simplified language model in the surface string level, it fails to deal with long distance dependency beyond a prescribed window size. Furthermore, because the source model make little use of generally available syntactic and semantic information in an MT system, it renders the source model to be order-free, and creates a huge number of possible translations, which would be limited if syntactic and semantic information is available. As a result, the parameter space increases exponentially with the window size. The limitations on the scope of the language model, in contrast to the large training corpus required and the large parameter space, are readily shown in [Brow 90]. Therefore, it might not be practical to adopt such a model for a system that has its own syntactic or semantic knowledge base already.

1.4. A Cooperative Approach

From a system designer's points of view, a statistical approach is suitable as an induction tool for acquiring a large and consistent set of fine-grained probabilistic "rules" [Su 90]. Since most linguistic knowledge is acquired by *induction* from various observations, many mechanical induction processes can be saved with an appropriate statistical model. Therefore, it is desirable to use a stochastic model for reducing the inconsistency inherent in manually constructed transfer rule base and reducing the labor in rule induction. However, it is not economic to adopt a purely statistical model, which requires a large parameter space due to the discard of syntactic and semantic information that is generally available in an MT system.

The parameter space can be reduced significantly if the transfer process can be modelled with syntactic and semantic features taken into consideration. For example, even in the context of an English-Chinese MT system, which deals with two drastically different languages, only about 4 or 5 local transfer operations on the various phrases are needed to perform the translation for a sentence of moderate length. This shows that structure information does provide a substantial amount of information, which can be used effectively for transfer. Our main concern is therefore to construct a statistical model on top of known and available syntactic and semantic information so that we can enjoy the advantages of both statistical models and general syntactic and semantic information.

In the following sections, a corpus-based approach that combines syntactic and semantic information with statistical mechanisms for transfer and generation is proposed. In particular, we

will focus on (1) the scoring mechanism for evaluating the preference of different transfer and generation possibilities, and (2) an algorithm for finding transfer units and transfer rules. The application of the proposed algorithm will provide substantial help in reducing the analysis results into a normalized form that is suitable for transfer. It also allows transfer rules to be acquired systematically, and allows generation rules to be acquired based on the target grammar. An annotated bilingual corpus with structure information is required to train the model. Preliminary results show that a consistent set of transfer rules, which consists of only a small number of transfer operations, can be acquired. Also, the transfer operations show very strong local properties, which make it possible to evaluate the preference of different transfer possibilities *via* decomposition of the source tree into local transfer units.

2. Translation Model and Translation Score

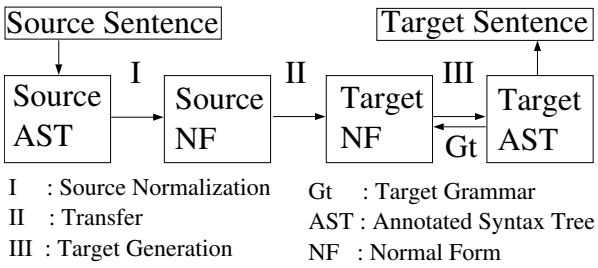

Figure 1 A conceptual model for the transfer and generation processes.

Figure 1 shows a model for the transfer and generation processes. In this conceptual model, the canonical semantic representation for the analysis phase is assumed to be an annotated syntax tree (AST) [Chan 92], which is a syntax tree annotated with syntactic and semantic features in each node. For simplicity in transfer and generation, it is desirable to introduce a normalized version, called the normal form (NF), of the AST, which consists of only *atomic* transfer units; this means that transfer operations can be accomplished simply by using local operations on such units. Under such a conceptual model, the source normalization process is conducted to reduce the source AST into its normal form. Source NF to target NF transfer is then performed, and the target NF is used to construct the target AST and the target sentence according to the target generation grammar (G_t).

Since there might be multiple choices in the transfer process, a scoring mechanism is required to rank different alternatives. According to [Chen 91], a transfer-based translation process can be modelled as an optimization process in which the best interpretation and transformation are searched to maximize a *translation score*. In an English-Chinese MT system, for instance, if $Cw_1^{n_C}, Ew_1^{n_E}, ELM, TM, CLM$ are the Chinese (i.e., target) sentence of length n_C, the English (i.e., source) sentence of length n_E, the English (source) language model, the transfer model, and the Chinese (target) language model, respectively, and I_C, I_E are the intermediate representations for the target and source sentences, then the *translation score* can be defined as follows:

$$
\begin{aligned}
& P(Cw_1^{n_C} \mid Ew_1^{n_E}, ELM, TM, CLM) \\
&= \sum_{I_C} \sum_{I_E} P(Cw_1^{n_C}, I_C, I_E \mid Ew_1^{n_E}, ELM, TM, CLM) \\
&= \sum \sum P(Cw_1^{n_C} \mid I_C, I_E, Ew_1^{n_E}, ELM, TM, CLM) \\
&\quad \times P(I_C \mid I_E, Ew_1^{n_E}, ELM, TM, CLM) \\
&\quad \times P(I_E \mid Ew_1^{n_E}, ELM, TM, CLM) \\
&\approx \sum \sum P(Cw_1^{n_C} \mid I_C, CLM) \qquad (generation\ score) \\
&\quad \times P(I_C \mid I_E, TM) \qquad (transfer\ score) \\
&\quad \times P(I_E \mid Ew_1^{n_E}, ELM) \qquad (analysis\ \ score) \\
&= \sum_{I_C} \sum_{I_E} P_G(.) \times P_T(.) \times P_A(.)
\end{aligned}
$$

In practical implementation, the searching strategy is to find the most probable analysis $I_{E\max}$ corresponding to $I_{Emax} \triangleq \arg\max_{I_E} P(I_E \mid EW_1^{n_E}, ELM)$ that maximizes the analysis score $P_A(\cdot)$ in the analysis phase; then search for the most probable intermediate representation $I_{C\max}$ for the Chinese sentence corresponding to $I_{Cmax} \triangleq \arg\max_{I_C} P(I_C \mid I_{E_{max}}, TM)$ that maximizes the *transfer score* $P_T(\cdot)$; then find the Chinese sentence $CW_{max} \triangleq \arg\max_{CW_1^{n_C}} P(CW_1^{n_C} \mid I_{C_{max}}, CLM)$ that maximizes the *generation score* $P_G(\cdot)$.

The analysis score or *score function* [Su 88, 91, Chan 92] is responsible for resolving ambiguity in the analysis phase. The transfer score and the generation score to be defined in the following sections account for the preference in the transfer and generation processes.

2.1. Transfer Score

The main task of the transfer phase is to bridge the gaps between the source and target structures. Since there might be multiple candidates for transfer, it is desirable to have a scoring mechanism to select the target construct that is most preferred. One example in English-Chinese translation is the preference of changing an English *noun* into its equivalent *verb* form in Chinese. For example, 'The use of NP ...' has several possible translations, all of which are acceptable. However, it would be more preferable in Chinese if its normal form is changed to the equivalent form of 'Using NP ...' or 'To use NP ...'. A careful analysis based on the distribution of the real translation examples would unveil the preference in the translation process. Therefore, it is desirable to have a *transfer* score for measuring the degree of preference for different stylistic and structural variations between source and target sentences.

However, the parameter space for the transfer score will be very large if the underlying transfer mechanism must be based on the tree-to-tree mapping from the source AST to the target AST. In this case, one might have to construct a large database for the various source-target AST pairs and their scores. On the contrary, the evaluation of the transfer score (and the generation score to be discussed later) will be much easier if each AST can be reduced to a normalized version, which can be decomposed into a set of atomic transfer units, called the *local transfer units* (LTU's), and the transfer and generation process can be accomplished simply by using *local operations*, such as local *permutation*, *insertion* or *deletion*, on such transfer units. Under such a circumstance, the transfer of each LTU will be independent of the other transfer units, and the transfer score can be evaluated easily *via* decomposition.

For instance, if I_C and I_E are the target and source normal forms which consist of sets of target LTU's X_i' and source LTU's X_i, respectively, and the local transfer units can be described with the $\overline{X}$ scheme [Sell 85], where each LTU consists of the syntactic category (X_i) for the unit, its head (h), arguments (a), specifiers (s) and modifiers (m) 4-tuples, and the associated features attached to these units, then the transfer score can be simplified, by decomposition, as follows:

$$
\begin{aligned}
&P\left(I_C \mid I_E, TM\right) \\
&= P\left(X_1', X_2', \cdots, X_n' \mid X_1, X_2, \cdots, X_n\right) \\
&\approx \prod_{i=1}^{n} P\left(X_i'\left(h', a', s', m'\right) \mid \alpha_i, X_i\left(h, a, s, m\right), \beta_i\right)
\end{aligned}
$$

where α_i, β_i are the left and right context of X_i. In deriving the last product term, we assume that X_i' depends on its corresponding LTU, X_i and its context within a limited window size only. When the context of the transfer units are ignored from consideration, the mapping will be a simple LTU transfer. This formula means to find the most probable target LTU's for the set of source LTU's, and this can be done by finding the best mapping between each LTU pair *via* decomposition. In general, the number of the LTU's is finite and has a reasonable size of several hundreds constrained by the syntactic structure; the required parameter space and training corpora can thus be kept reasonably small.

Note that the syntactic and semantic roles of the LTU and its 4–tuple can be changed during transfer. Nodes can also be inserted or deleted as desired. This formulation thus provides sufficient expressive power for the transfer process and enables drastic stylistic and structural variations between the language pair to be handled.

Figure 2 shows an example of normal form for the above mentioned formulation. In fact, any appropriate head-argument structure that satisfies certain good local transfer properties can be used as an LTU. An algorithm will be outlined later to find possible candidates for such units automatically from a bilingual corpus.

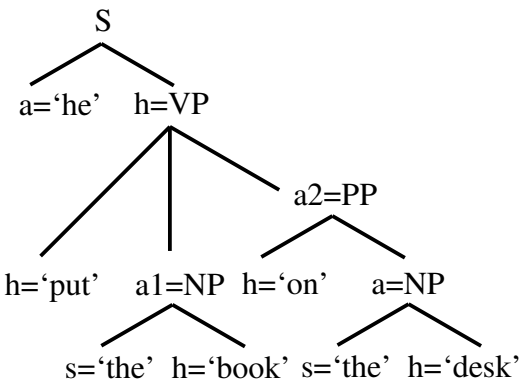

Figure 2 The normal form for the sentence 'he put the book on the desk'.

2.2. Generation Score

To generate the target sentence from the target normal form, one has to conduct *structure generation* for reordering the selected target lexicon and perform *morphological generation* for generating target-specific morphemes, particles, and so on. For instance, the Chinese-specific words 'ba', 'bei' (passive voice marker), and classifiers for noun must be generated during the morphological generation phase. Different sentence patterns are preferred differently in different situations. A generation score is thus required to measure the preference for the various possible sentence patterns.

Let the target sentence $Cw_1^{n_C}$ be represented by a set of sentence segments t_i, a permutation be specified by a permutation function $u\left(X_i'\right)$ and the (morphological) generation be represented by a generation frame $f\left(X_i'\right)$ for the LTU X_i', then the generation score can be formulated as follows, based on the assumption of local transfer:

$$
\begin{aligned}
& P(Cw_1^{n_C} \mid I_C, CLM) \\
& \approx P\left(t_1, t_2, \cdots, t_n \mid X_1', X_2', \cdots, X_n'\right) \\
& \approx \prod_i P\left(f\left(X_i'\right), u\left(X_i'\right) \mid \alpha_i', X_i'\left(h', a', s', m'\right), \beta_i'\right) \\
& = \prod_i P\left(u\left(X_i'\right) \mid \alpha_i', X_i'\left(h', a', s', m'\right), \beta_i'\right) \\
& \times P\left(f\left(X_i'\right) \mid u\left(X_i'\right), \alpha_i', X_i'\left(h', a', s', m'\right), \beta_i'\right).
\end{aligned}
$$

where α_i', β_i' are the left and right context of X_i'. The best target sentence that maximizes the generation score will be selected from the various possible sentence patterns. As usual, we assume that the i-th sentence segment depends only on the i-th LTU. Note that the first product term in the last formula corresponds to the structure generation (or structure transfer) process, and the second term corresponds to the morphological generation phase.

The permutation function specifies the relative positions of the lexicons for the 4–tuple. The generation frame, on the other hand, specifies which target-specific morphemes are to be inserted to the particular positions in between the elements of the 4–tuple. For example, when a normalized verb phrase

$$
\begin{aligned}
& VP(h, a1, a2, s, m) \\
& = VP(put, the-book, on-the-desk, NIL, NIL)
\end{aligned}
$$

is used to generate its Chinese equivalent

$$
VP(BA, book, put, on-desk),
$$

the permutation function would be $u(h, a1, a2, s, m) = (a1, h, a2, s, m)$ and the generation frame would be $f(a1, h, a2, s, m) = (BA, a1, \emptyset, h, \emptyset, a2, \emptyset, s, \emptyset, m, \emptyset)$ for the Chinese particle 'ba' ('$\emptyset$' is a null symbol).

3. Learning Transfer Units and Operations

One key factor to the success of the transfer and generation model depends on the existence of the localized transfer units and a consistent set of transfer operations ("transfer rules") governing the transfer process. Such units will render the computation of transfer or generation score an easy task through *decomposition* of the syntax tree or normal form. Practically, one may not find an ideal underlying transfer mechanism and conduct transfer by using only local operations. However, such transfer units do exist for many constructs. (The worst case is to use the whole sentence as a transfer unit.) To define a normal form that has desirable local transfer properties, we can apply the following "index permutation" algorithm to a bilingual tree bank to find the candidate transfer units and the corresponding transfer rules. According to the model in Figure 1, the processes to be learned are the transfer processes (i) between the source AST and the source normal form, (ii) between the source-target normal form pairs, and (iii) between the target normal form and the target AST. However, for simplicity of illustration, we will show how to find the candidates of LTU's and the corresponding transfer rules from the source syntax tree and its corresponding target translation. The learning algorithm for the other tree pairs is essentially identical and, is , in fact, easier.

Now, refer to Figure 3 for a source AST and target translation pair. (T_1, T_2, ... are the equivalent target words for the source words S_1, S_2, ..., respectively.) First, the corresponding tokens in the source sentence and the target sentence need to be aligned. The alignment can be done automatically or semi-automatically by using the transfer dictionary. For simplicity, first assume that the terminal words of the source syntax tree have a one-to-one correspondence with the terminal words of the target sentence. (We will deal with the case that is not 1–1 correspondent later.) Each word in the *target* sentence is assigned a word index in ascending order as shown

at the bottom of Figure 3. The same index is also passed to its equivalent word in the source sentence. Each node in the source syntax tree is then associated with an *index list*. The index list for a terminal node is its word index. The index list for a nonterminal node is acquired by concatenating the index lists of its children nodes in a left-to-right manner. Figure 3 shows an example of a syntax tree whose nodes are labeled with index lists.

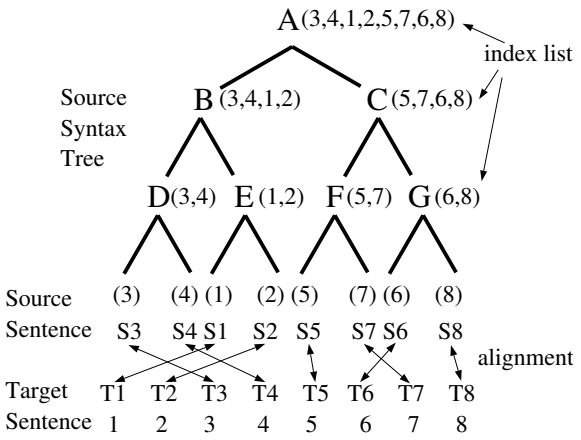

Figure 3 Syntax tree with attached word index list.

Note that if a group of source words are translated, as one unit, into a group of consecutive target words without reordering, then the common ancestor node of these source words must have a sequence of consecutive word indices in its index list. The index lists thus contain reordering information of the transfer process. The nodes of a source syntax tree can be classified, according to their index lists, into 3 types: (1) the index list is a sequence of *consecutive* and *sorted* integers (e.g, D(3,4), E(1,2)); (2) the word indices are *consecutive* but *not sorted* (e.g., B(3,4,1,2), C(5,7,6,8), A(3,4,1,2,5,7,6,8); (3) the word indices are still *not* consecutive integers even after sorting (e.g., F(5,7), G(6,8)). Different node types require different transfer operations to make their indices consecutive and sorted just like their target counterparts.

If a node belongs to *type-1*, then all the terminal strings rooted at this node must have been translated to a sequence of consecutive target words in the target sentence without changing their linear order. Such a node requires NO OPERATION (NOP) to change its local structure during transfer. For instance, node D is a type-1 node, and its children (S_3, S_4) are translated into (T_3, T_4) without changing their relative linear order in the target sentence.

A *type-2* node is a possible candidate of local transfer unit. The terminal strings rooted at such a node are translated to a sequence of consecutive target words as suggested by the index list; the fact that the index list is not sorted further implies that its children are reordered during transfer. Furthermore, the operation required to change the linear order of the source words to the order of their target equivalent can be acquired easily by inspecting the relative positions of the index lists of its children. For instance, the terminal strings (S_3, S_4, S_1, S_2) rooted at node B can be transferred into their equivalent target words (T_1, T_2, T_3, T_4) by the following transfer operation:

$$
\begin{aligned}
B\,(3,4,1,2) &\rightarrow D\,(3,4)\;\; E\,(1,2) \\
&\Downarrow local\;\; transfer \\
B\,(1,2,3,4) &\rightarrow E\,(1,2)\;\; D\,(3,4)
\end{aligned}
$$

The required transformation corresponds to a permutation of the children which makes the word indices of the children appear in ascending order. We will call such a transfer operation an XCHG (EXCHANGE or permutation) operation.

Finally, if a node is of *type-3*, the non-consecutive word indices imply that the source words rooted at such a node must be translated to non-consecutive positions; their target words must be intermixed with other target words. For example, the children of the type-3 node F(5,7) is translated to the 5th and 7th positions. To be locally transferable, a type-3 node must be merged (MERGE) with other nodes so that their children can be reordered by way of local permutation. Note also that if such nodes are merged to form a unit that can be reordered by local permutation, then the merged node must be of type-2. Therefore, all type-3 nodes that have a common type-2 ancestor node can be merged to form a transfer unit. To keep the number of children of the merged node as small as possible, one should stop the merge operations at the nearest 'common ancestor'. The merged *type-3* nodes will also form a candidate of local transfer unit. For example, the type-3 nodes F(5,7) and G(6,8) can be merged to form a new node, say $C^*(5,7,6,8) \rightarrow S_5\ S_7\ S_6\ S_8$. An XCHG operation can be performed afterward to generate the correct order.

The above algorithm can be extended when the source words and target words are not one-to-one correspondent. (1) If a source word corresponds to more than one target words, the source word can be associated with an index list consisting of all the indices of the target words. (2) If more than one source words are translated into one target word, then the source words can be associated with the word index of the single target word. The algorithm then proceeds as in the 1–to-1 instances for the above two cases. (3) If a source word is mapped to zero target word, the source word can be associated with a null word index. The mother node of such a node requires a DELETION (DEL) operation to delete the source word. (4) On the contrary, if a *target* token does not have a corresponding source word, then at least one type-3 node (with that target word index missing) will appear in the source syntax tree. In this case, an INSERTION (INS) operation is required to insert a pseudo node to the type-3 node that is closest to the source terminal words. (5) In some special cases, a source word may have to be replaced with another equivalent source word before aligning it with the target word. This situation occurs, for instance, when an English pronoun is translated into the Chinese word corresponding to its referent. Other drastic changes in syntactic or semantic features also requires a similar manipulation. Such a transfer operation will be called a REPL (REPLACEMENT) operation.

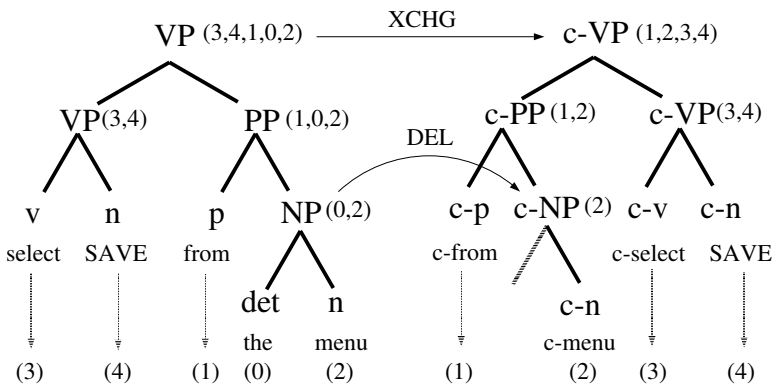

Figure 4 An example of English-Chinese transfer process.

Figure 4 shows the transfer operations involved in translating the English verb phrase "... select SAVE from the menu" into its Chinese counterpart. All symbols of the form 'c-xxx' represent the Chinese equivalent of the English symbol 'xxx'. The index '0' is shown explicitly in the source syntax tree to show the position of the missing determiner. As will be shown later, most transfer operations and transfer rules can be described in terms of the NOP, INS, DEL, XCHG, REPL and MERGE operations (or a composition of them) in a mechanical and consistent way. Furthermore, a source tree node and a target tree node with the same index list will form a translation pair. For instance, if a *target* tree node B' exists such that $B' \stackrel{*}{\Rightarrow} T_1T_2T_3T_4$, then the subtree rooted at

$B'(1,2,3,4)$ is the counterpart of B(3,4,1,2) (since $B \stackrel{*}{\Rightarrow} S_3 S_4 S_1 S_2$), and vice versa. Such translation pairs, like B and B′, thus provide good candidates for us to define the normal forms. It is possible that the transfer operations like INS and MERGE be applied ambiguously at different nodes yet produce the same target strings. In this case, it is desirable to attach the nodes in a way that make the source tree and the target tree most similar. Such a strategy might reduce the possibility of further nonlocal operations. A feasible similarity measure is the number of matched translation pairs.

By visiting the tree in a bottom-up manner, one can find the minimal number of transfer operations required to transfer a tree into its target equivalent and find the candidates for local transfer units. These units will provide useful information in defining the normal forms. If the normal forms are not to be bounded strictly by syntax, the LTU candidates acquired with the above algorithm can be further reduced to a deeper structure. The reduction process will involve examining the features associated with these candidates and making appropriate adjustment.

4. Learning Target Generation Rules

The above algorithm can also be applied to the tree to normal form mapping and normal form to normal form mapping because they are just different types of trees in essence. When applying such an algorithm between the *target normal form* and the *target AST* in the reverse direction, we can acquire a set of generation rules based on the *target* grammar. The generation rules acquired in this way differ significantly from their counterpart for a conventional transfer-based system. In particular, the transfer operations between the target normal form and the target syntax tree are learned from the annotated syntax trees of the *target* language, which is acquired by parsing the target sentences with the analysis grammar of the target language. Hence, the corresponding generation grammar is exactly the analysis grammar of the target language, not a grammar adapted from the source grammar through the one-way "analysis-transfer-generation" process; the generation process using such generation rules is simply a reverse process of analyzing the target sentence with the target grammar. Hence, the generated sentence will strictly follow the grammar and style of the target language.

The above algorithm thus provides an important mechanism for producing high quality translation. This is not possible for a generation grammar that is adapted from the source language side; in this case, the style of the generated sentence will be bounded by the *source* grammar, which might not be natural to the native speakers of the target language. To produce high quality translation, we can thus adopt a *bidirectional* strategy in which the required transfer rules are trained from the source AST and the source normal form, and the generation rules are learned from the target AST (acquired by parsing the target sentence) and the target normal form [Su 92b]. In the meantime, the transfer score and the generation score can be tuned to make the best selection.

5. Experiments

To show the superior properties of the above model, preliminary experiments are conducted on a small bilingual corpus. The small corpus contains 111 English-Chinese sentence pairs and their corresponding parse trees, which are generated by the BehaviorTran English-Chinese MT system. (The average sentence length for the source (English) language is 13 words.)

The source trees contains 6126 nodes on which local operations can be conducted. However, only 564 transfer operations, as shown in Table 1, are required to acquire the target translation of all the sentences. This corresponds to about 5.1 operations per sentence. Such observation justifies our previous assumption that structure information can provide great reduction in searching for feasible transfer operations in contrast to a purely statistical model where the alignment patterns grow exponentially with the word length.

Operation	Frequency	Percentage
DEL	230	40.8%
INS	158	28.0%
XCHG	122	21.6%
MERGE	50	8.9%
REPL	4	0.7%

Table 1 Distribution of Transfer Operations

Among the transfer operations, only the MERGE operations require nonlocal movement. Fortunately, the nonlocal operations constitute only about 8.9% of the transfer operations. Also, the merged nodes are found to be less than 2 tree levels apart in most cases. The transfer operations can thus be conducted quite locally, and the assumption in reducing the transfer score and generation score based on the local transfer units is quite reasonable.

Table 2 shows a few typical transfer operations captured by the proposed algorithm. The transfer operations are shown in the "OP" column and their frequency of occurrence is shown in the "Freq" field. The "Transfer Unit" field shows the node on which the transfer operation is conducted, and the final column shows the new node configuration after the transfer operation is conducted. Deleted or inserted constituents are quoted in an angle bracket pair ("<>"). A special composite transfer operation "XI" (XCHG plus INS), which tends to occur at the same time for a special construct, is also shown.

Freq	OP	Transfer Unit	Configuration after Transfer
114	DEL	N2 -> <DET> N1	=> N1
42	DEL	P1 -> <P*> N3	=> N3
16	DEL	V3 -> <AUX> V2	=> V2
15	DEL	N2 -> <DET> NLM* N1	=> NLM* N1
10	DEL	NMOF -> <CMPR> SI-2A	=> SI-2A
20	XI(+)	N3-A -> N2 NMF*	=> NMF* <DE> N2
16	INS	SDEC -> N3 V4	=> N3 V4 <DE>
15	INS	S2 -> SAC* S	=> SAC* <please> S
10	INS	N2 -> DET N1	=> DET <CL> N1
8	INS	P1 -> P* N3	=> P* N3 <LOC>
75	XCHG	N3-A -> N2 NMF*	=> NMF* N2
19	XCHG	V2A -> V1 ADV*	=> ADV* V1
5	XCHG	SDEC -> N3 V4	=> V4 N3
(+) XI: XCHG and INS			

Table 2 Typical Transfer Operations in English-Chinese machine translation

The first five DEL operations, for instance, show that the English article or determiner (DET) in a noun phrase (N2), some prepositions (P*) in a prepositional phrase (P1), the verb "be" and some auxiliary verbs (AUX) before a verb phrase (V2), the complementizer or relative pronoun (CMPR) in a post-nominal modifier (NMOF) containing incomplete sentences (SI-2A) tend to be deleted in English-Chinese translation. The INS operations correspond mostly to the insertion of the Chinese "de" particle (<DE>), classifier (<CL>), locative morphemes (<LOC>), the English equivalent of "please" (<please>), and so on. The XCHG operations correspond roughly to the structural transfer process. They are conducted to change a group of the post-nominal modifiers (NMF*) or a group of verb modifiers (ADV*) of the English sentences to the front of the noun phrases or verb phrases (V1). They are also conducted to change the passive voice in English declarative sentences (SDEC) into the active voice expression in Chinese sentences. As a result, the subject noun phrases (N3) are swapped to the rear of the verb phrases (V4).

A closer look at the transfer operations in the acquired transfer rules shows that more than 75% of the transfer operations are covered by the top 24 transfer operations. From the brief summary in Table 2, it is obvious that the captured transfer operations agree with the general linguistic observations quite well. The proposed model thus provides a favorable way in modelling the transfer process and acquiring the transfer rules in a systematic and consistent way.

For the present tests, most of the transfer operations can be described simply in terms of syntactic features, and the syntax trees are used to play the role of a normal form directly. For a larger domain, it is expected that the more semantics-oriented normal forms as described previously will be needed in order to deal with more critical transfer operations and more syntactic variants. Nevertheless, the training procedure with the normal form trees is essentially identical to the procedure with syntax trees. The proposed model can thus be adopted without modification. The related topics at a semantic level are now under investigation and testing.

6. Conclusions

In this paper, a transfer score and a generation score are proposed as the preference measures for the transfer and generation processes. The corpus-based approach based on such formulation enables us to evaluate different preferences for the various possibilities of transfer and generation. An algorithm is proposed to acquire a set of local transfer units and transfer rules for a language pair. This algorithm enables one to accomplish the transfer process and the generation process by using local transfer operations. Furthermore, by applying the algorithm to learn the generation rules from the target language directly, the generation phase can produce target sentences that strictly follow the grammar and the style of the target language. The system can thus be tuned to produce target sentences according to the target grammar, instead of a modified version of the source grammar.

References

[Benn 85] Bennett, W.S. and J. Slocum, 1985. "The LRC Machine Translation System," *Computational Linguistics*, vol. 11, no. 2–3, pp. 111–119. 1985.

[Brow 90] Brown, P. F., et al. 1990. "A Statistical Approach to Machine Translation," *Computational Linguistics*, vol 16, no. 2, pp. 79–85. 1990.

[Chan 92] Chang, J.-S., Y.-F. Luo and K.-Y. Su, 1992. "GPSM: A Generalized Probabilistic Semantic Model for Ambiguity Resolution," *Proceedings of ACL-92*, pp. 177–184, 30th Annual Meeting of the Association for Computational Linguistics, University of Delaware, Newark, DE, USA, 28 June–2 July, 1992.

[Chen 91] Chen, S.-C., J.-S. Chang, J.-N. Wang, and K.-Y. Su, 1991. "ArchTran: A Corpus-Based Statistics-Oriented English-Chinese Machine Translation System," *Proceedings of Machine Translation Summit III*, pp. 33–40, Washington, D.C., USA, July 1–4, 1991.

[Naga 84] Nagao, M. 1984. *A framework of a mechanical translation between Japanese and English by analogy principle*, In. A. Elithorn and R. Banerji (eds.) Artificial and Human Intelligence, pp. 173–180, North-Holland, Amsterdam.

[Naga 85] Nagao, M., J. Tsujii, and J. Nakamura, 1985. "The Japanese Government Project for Machine Translation," *Computational Linguistics*, vol. 11, no. 2–3, pp. 91–110. 1985.

[Sato 90] Sato S. and M. Nagao, 1990. "Toward memory-based translation," *Proceedings of COLING-90*, vol. 3, pp. 247–252, Helsinki, Finland.

[Sell 85] Sells, P., 1985. *Lectures On Contemporary Syntactic Theories,* CSLI Lecture Notes Number 3, Center for the Study of Language and Information, Leland Stanford Junior University., 1985.

[Su 88] Su, K.-Y. and J.-S. Chang, 1988. "Semantic and Syntactic Aspects of Score Function," *Proc. of COLING-88,* vol. 2, pp. 642–644, 12th Int. Conf. on Computational Linguistics, Budapest, Hungary, August 22–27, 1988.

[Su 90] Su, K.-Y., and J.-S Chang, 1990. "Some Key Issues in Designing MT Systems," *Machine Translation*, vol. 5, no. 4, pp. 265-300, 1990.

[Su 91] Su, K.-Y., J.-N. Wang, M.-H. Su and J.-S. Chang, 1991. "GLR Parsing with Scoring," In M. Tomita (ed.), *Generalized LR Parsing*, Chapter 7, pp. 93-112, Kluwer Academic Publishers, 1991.

[Su 92a] Su, K.-Y and J.-S. Chang, "Why Corpus-Based Statistics-Oriented Machine Translation," *Proceedings of TMI-92*, pp. 249–262, 4th Int. Conf. on Theoretical and Methodological Issues in Machine Translation, Montreal, Canada, June 25–27, 1992.

[Su 92b] Su, K.-Y., M.-W. Wu and J.-S. Chang, "A New Quantitative Quality Measure for Machine Translation Systems," *Proceedings of COLING-92*, vol. II, pp. 433–439, 14th Int. Conference on Computational Linguistics, Nantes, France, July 23–28, 1992.

[Su 93] Su, K.-Y. and J.-S. Chang, "Why MT Systems Are Still Not Widely Used?" *Machine Translation*, vol. 7, no. 4, pp. 285–291, Kluwer Academic Publishers, 1993.

[Tsut 90] Tsutsumi, T. 1990. "Wide-Range Restructuring of Intermediate Representation in Machine Translation," *Computational Linguistics*, vol. 16, no. 2, pp. 71–78, June 1990.