A Corpus-Based Statistics-Oriented Two-Way Design for Parameterized MT Systems: Rationale, Architecture and Training Issues

*Keh-Yih Su, *Jing-Shin Chang and +Yu-Ling Una Hsu

*Department of Electrical Engineering
National Tsing-Hua University
Hsinchu, Taiwan 30043, R.O.C.

+Behavior Design Corporation
No. 5, 2F, Industrial East Road IV, Science-Based Industrial Park
Hsinchu, Taiwan 30077, R.O.C.

Email: *kysu@bdc.com.tw, *shin@hermes.ee.nthu.edu.tw, +una@bdc.com.tw

ABSTRACT

In many conventional MT systems, the translation output of a machine translation system is strongly affected by the sentence patterns of the source language due to the one-way processing steps from analysis to transfer and then to generation, which tends to produce literal translation that is not natural to the native speakers. The literal translation, however, is usually not suitable for direct publication to the public unless a great deal of post-editing efforts is made. In this paper, we will propose a training paradigm for acquiring the transfer and translation knowledge in a corpus-based statistics-oriented (CBSO) parameterized MT system from a bilingual corpus with a two-way training method. In such a training paradigm, the knowledge is acquired from both the source sentences and the target sentences. It is thus possible to avoid the translated output from being affected by the source sentence patterns. Training methods for adapting the parameter set to the various specific user styles are also suggested for the particular needs in restricted domains. Because it provides a flexible way to adapt the system to the various domains (or sublanguages), it is expected to be a promising paradigm for producing high quality translation according to user preferred styles.

1. Introduction

Most traditional MT systems acquired their underlying knowledge with a one-way design paradigm, in which the underlying translation knowledge, in particular, the transfer knowledge, is largely derived based on the source sentences; as a result, the output translation is often strongly affected by the sentence patterns of the source language, and the output is usually too literal [Somers 93, Su 93]. In addition, the system modules are mostly not in a parameterized form; consequently, it is hard to tune the system for a specific domain and thus unable to produce high quality translation in the subdomain, which is required in practical applications. In this paper, we will indicate the possibility for removing such source dependency under a parameterized MT architecture by using a two-way training method to acquire the translation and transfer knowledge automatically from a bilingual corpus. An MT system with output not biased by the source sentence pattern can thus be constructed with small cost.

Another advantage of training the system parameters (i.e., the probabilities in our case) from a

July 5-7, 1995
Centre for Computational Linguistics
Katholieke Universiteit Leuven
Leuven, Belgium

bilingual corpus is that it is possible to cofigure an MT system to fit the various styles in different applications with different sets of parameters. Such an approach, in contrast to the conventional analysis-transfer-generation architecture, make it possible for adapting itself to user's feedback by adjusting those parameters.

To make an MT system parameterizable, the underlying language models have to be expressed in a form that is linguistically appropriate and computationally suitable for preference evaluation; and a set of quantitative formulations has to be developed for finding the most preferred candidate for the various analysis, transfer and generation phases. In this paper, a framework will be proposed for such a parameterization task through several different levels of normalization.

It is also important that the parameters of a parameterized system could be easily acquired or trained. In this paper, special attention will be paid to construct such a system with a small annotated corpus. We will characterize such a training process as a two-end constraint optimization problem to reflect its attempts for two-way training from a bilingual corpus. Under such an optimization approach, the system designer could concentrate on the acquisition or compilation of lexicon knowledge and the tagging of shallow syntactic and semantic features of the lexicons.

2. Why Parameterized MT Systems ?

In many real world applications, the MT users or customers, like large international computer companies, do require that the MT system produces "publishable" translation (say, for computer manuals), instead of only readable or understandable messages (which might be suitable for information retrieval applications). Besides, such companies may have their own long established publishing style, which must be considered in order to win their contracts [Su 93]. A practical MT system, therefore, needs the capability to produce high quality translation and adapt itself to a user-specific style.

High quality translation, however, in general, is possible only in a very restricted domain or for some sublanguages; it is hard for a practical MT system to have a wide coverage in applications yet retaining high quality translation. To produce the best possible output, most MT systems, therefore, would operate in a sublanguage and domain specific manner. (Even for a restricted domain, it is not easy to acquire the required fine-grained knowledge for producing fluent translation in the target language.) However, to operate an MT system economically, the amount of translation must be large enough, which means that an MT system must be able to operate in several domains at the same time. Therefore, a wide coverage is still a critical issue for a practical MT system. For these reasons, adapting an MT system to the various domains and producing publishable translation under such domains is thus probably the most promising strategy for operating a practical system.

To construct such a system, it is important to consider the following issues: (1) what knowledge to use to produce fluent translation for a restricted domain, (2) how to acquire such knowledge, (3) how to adapt the output style to a particular customer, and (4) how to incorporate post-editor feedback on-line to avoid repeated errors and complaints from the post-editors.

Traditional MT systems rely heavily on a large number of rules to practice the language processing needs. For instance, the major knowledge sources for a transfer-based MT system would include lexicon information, analysis grammar, transfer rules, generation grammar and the rules for disambiguation and semantic interpretation. Many such works are completed with huge human efforts. Therefore, the cost required to develop such a system with fine-grained knowledge is usually high in comparison to other software systems. It is therefore not feasible to construct different sets of rules for different applications due to the high cost. And it is not easy to keep them consistent across the various releases (or generations) of the system. Consequently, most rule-based MT's either operate in general domains but generate relatively poor output, or generate high quality translation in only one or two very restricted domains (e.g., the METEO system [Hutchins 86]). Such MT systems are, therefore, hard to get enough pay-back to support further research. To relieve the burden in knowledge acquisition above-mentioned, symbolic learning methods had also been tried to organize the linguistic knowledge. However, such approaches are usually awkward in handling uncertain knowledge and do not gain much success so far.

In contrast to the conventional systems, a parameterized system architecture maybe a more practical solution. A parameterized system is characterized by a quantitative optimization criterion and a training mechanism for acquiring the language parameters (such as a set of probabilities or scores) from real text corpora. The training mechanism is simply an estimation process to get the parameter sets from a corpus according to some objective optimization criteria. Alternatively, the parameter sets could also be adjusted to reflect user preference under other optimization criteria; such a system could be characterized as a feedback-controlled parameterized MT system [Su 92b]. In contrast to other systems whose knowledge is either transformed from existing linguistics theory or acquired from symbolic learning methods, a parameterized system as characterized here could have the following potential advantages.

First of all, parameter learning is usually easier and more objectively optimized than symbolic learning approaches, in which the underlying rules may not handle uncertainty or preference objectively. A parameter learning process usually involves only mathematical computation instead of complicated induction mechanisms. Therefore, the driving mechanism is simple and each learning step could be quantitatively controlled. Furthermore, the search path toward the best parameter set could be found easily by observing the dependency of the optimization criterion as a function of the parameters in the parameter space, and adjusting the parameters in the direction that is most likely to increase the value of the optimization function [Amari 67].

Secondly, the parameter sets could be easily adjusted for the various styles in different domains in a systematic way. The knowledge acquisition cost, in terms of man-years, is usually smaller. A parameterized system thus provides the potential benefits of using alternative sets of parameters for different applications, and leaves the driving mechanism, functional modules (or the knowledge base) the same. Therefore, a parameterized system is preferred in terms of knowledge acquisition cost and adaptability to different requirements.

A parameterized system could usually be constructed by first properly normalizing the intermediate representations of the various modules and then extracting useful features from such normalized constructs. The architecture in Figure 1 shows such a parameterized system, where PT represents the parse tree, NF1 represents the first-level normal form (referred to as the 'normalized syntax tree'), NF2 represents the second-level normal form (or 'semantic tree') of a source (S) or target (T) sentence; the subscripts 's' and 't' stand for the source and target sentences respectively. P(X|Y) represents the conditional probability for X to appear given that Y is observed. Such parameters (conditional probabilities) are used to assign preference scores for disambiguation. (See the next section for more details.)

We further assume that the parse trees are produced based on a phrase structure grammar G, the NF1 constructs are produced based on a set of normalization rules, NR1, and the NF2's are produced according to a second set of normalization rules, NR2. In addition, the reverse operations are directed by sets of generation rules of the various levels (GR2, GR1, and GR0), which specify the sets of possible $NF1_t$, PT_t and T's that could be enumerated in the reverse direction of the analysis processes. In this flow, a "normal form" refers to a properly normalized intermediate representation. (Whenever appropriate, the term "normal form" will also include the parse trees in the following discussion.)

Note that, in such a system, the phrase structure grammars, the normalization rules and the generation rules only produce possible parses or normalized constructs without involving in the disambiguation tasks; the system parameters (i.e., the conditional probabilities), on the other hand, play the major roles for disambiguation or selection of preferred constructs based on quantitative preference scores.

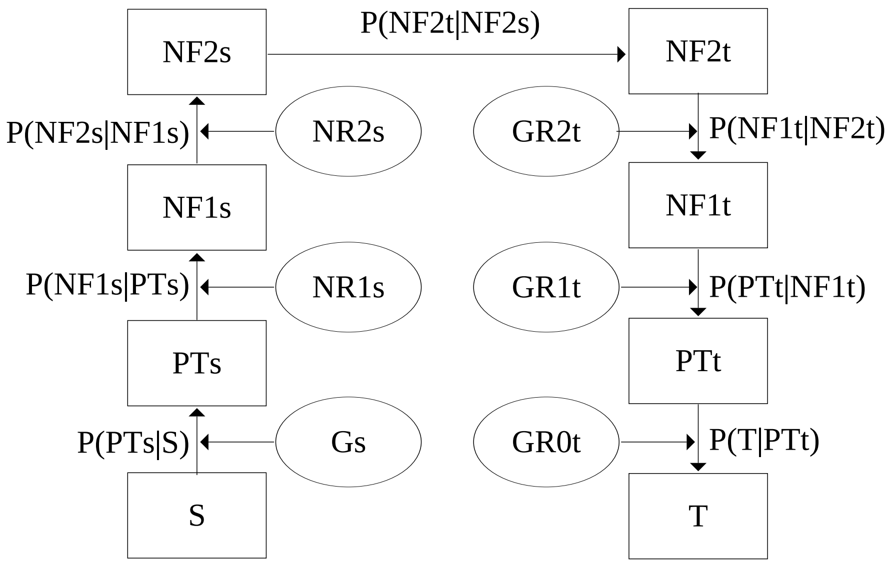

Figure 1 Translation Flow for a Parameterized MT System

3. Two-Way Design vs. One-Way Design

As mentioned above, the target of an MT system is to produce fluent output that is natural to the native speakers of the target language. Under traditional transfer-based MT architecture, however, most output translations are strongly influenced by the sentence patterns of the source language and many literal translations are produced across the transfer phase [Somers 93, Su 93]. Such source-dependency is easily introduced to a transfer-based MT system in the stratified analysis, transfer and generation phases as described below.

First, in the transfer phase, the transferred lexicon or structure for the target language is usually a locally modified version of the corresponding source lexicon and structure, because many designers tend to use minimal numbers of local adjustments to get reasonable translations. As a result, the mapping may retain a large portion of the source information, such as the sentence pattern. The mapping, therefore, may minimize the required transfer operations, but may not optimize the translation quality; the output is usually only readable and is too literal for a native speaker. And, the target sentence generated may not fall within the range of the sentences produced by a native post-editor.

Secondly, the designers usually design the transfer rules explicitly or implicitly based on the source training corpora. The transfer rules derived thus can produce readable translations for the training corpora. Such transfer rules, however, may not be fired correctly for the unseen sentences and thus produce illegal sentences.

Finally, even though the target sentence generated is OK when judged sentence by sentence by the native speaker, it may not be in a preferred style under the discourse context, because most sentences are generated on a sentence-by-sentence basis and the generation grammar may not take the target text around the current sentence into consideration.

Consequently, a traditional transfer-based MT system, in the transfer phase, may prefer a target construct that is biased by the source sentence pattern. In the generation phase, such style is inherited; a literal translation that is strongly influenced by the source sentence patterns is thus produced. In addition to source dependent, it is also hard to make the system reversable. Therefore, many target language information and modules may not be reused when it becomes the source language and vice versa.

A system designed in this way will be characterized as a one-way design system. By one-way, we mean that almost all the translation knowledge is derived, explicitly or implicitly, based on the training corpus of the source language. A system that is characterized as a one-way system will not be able to learn proper transfer knowledge that governs the translation of the source sentences and the most preferred target sentences. The underlying transfer knowledge acquired in this way thus may not be able to generate the most preferred target sentences.

Many transfer-based system may fall within this category due to the inherent operation of the stepwise analysis, transfer and generation flow. Of course, a transfer-based system could be made a two-way system by changing the knowledge acquisition procedure with a bilingual corpus. The famous statistical MT framework in [Brown 90] could be regarded as a two-way system with the above characterization since the translation knowledge is trained from a bilingual corpus. However, because it trains the translation knowledge in the surface string level, and does not use any normalized construct at a higher syntactic or semantic level, it has some disadvantages in implementing a large scale system [Su 92a, Chang 93]. Notably, the parameter space for such a system will be extremely large, where some of the parameters may be used simply to learn some of the known syntactic or semantic constructs. Furthermore, it lacks the capability in dealing with long distance dependency beyond a small window size.

To change the system architecture from one-way design toward two-way design, the transfer knowledge should thus be trained from both properly normalized source and target knowledge representations, which should both fall within the range of the sentences that will be produced by the native post-editors, according to the discourse context of the source language and target language respectively. The following flow shows the general idea for training a two-way system. The bold arrows at the right hand side emphasize that the intermediate representations for the target language are directly derived from the target sentences in an aligned bilingual corpus.

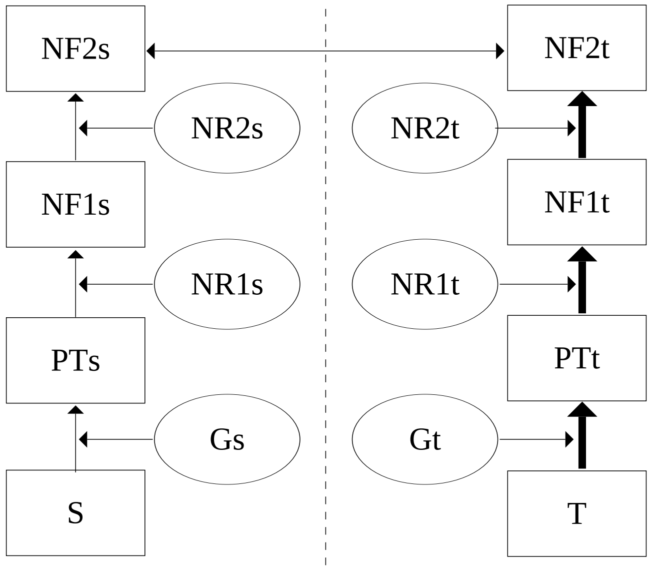

Figure 2 Two-Way Training Flow for a Parameterized MT System

Note that, the translation flow still follows the analysis, transfer and generation steps, but the training procedure for knowledge acquisition is different from the one-way design system. The arrow symbols indicate that the PT's, NF1's and NF2's for *both* the source and target sentences are derived from the source and target sentences respectively, based on their own phrase structure grammars and normalization rules. Therefore, all such intermediate representations are guaranteed to fall within the range of the sentences that will be produced by the native speakers of the source and target languages; the transfer phase only *select* those preferred candidates among such constructs. In addition, the transfer parameters are estimated based on such intermediate representations and the transfer knowledge is derived from both the source and target sentences of an aligned bilingual corpus. More detailed information will be given in section 6.

In particular, (1) the target normal forms are directly derived from the target grammar, not a deformed version from the source grammar, and (2) the mapping between the source and target normal forms are tuned to generate the sentences which reflect the preferred style of a particular user.

4. Architecture of a Parameterized MT System

The various steps of normalization are required to reduce the number of parameters required to characterize the whole parameterized MT system. The following sections show an example of such normal forms in an undergoing project of the BehaviorTran bidirectional MT system [Su 90, Chen 91, Chang 93, Hsu 94]. An architecture that implements the above parameterized bidirectional MT system could be characterized with the following modules.

4.1 Syntactic Parsing

The first module is a syntactic parsing module, including morphological processing, which produces the parse tree (PT) of the input sentence according to the phrase structure grammar of the source language. This module provides the syntactic information for the input sentences. An example syntax tree is shown in the following figure. (‘‘To meet spectrum analyzer specifications, allow a 30 minutes warm-up before attempting to make any calibrated measurements .’’)

```
 SJ
  |
SIMP1 --------------------------
  |                             \
 ADTZ ----------------------      SIMP -----------------------
  |                         \      |                          \
 ADT                          \    V1                           ADTC
  |                            |   |  \                          |
 SBJ                           |   VJ   VN-T                    SBJ
  |                            |   |     |                       |
  SB                           |   |    N3J                      SB  --
  |                            |   |     |                       |     \
 SIJ                           |   |   N3-AJ                     |      V2-AJ
  |                            |   |     |                       |        |
  SI                           |   |    N3-A                     |       V2-A
  |  \                         |   |     |                       |        |
  |    V2J                     |   |    N2J                      |        V1
  |     |                      |   |     |                       |        |  \
  |     V2                     |   |     N2                      |        VJ  VSI-T
  |     |                      |   |     |  \                    |        |     |
  |     V1  -                  |   |    DET    N1                |        |    SIJ
  |     |    \                 |   |     |     |                 |        |     |
  |     VJ     VN-T            |   |     |     N*  ---           |        |     SI
  |     |       |              |   |     |     |      \          |        |     |  \
  |     |      N3J             |   |     |     N*       \        |        |     |   V2J
  |     |       |              |   |     |     |         |       |        |     |    |
  |     |     N3-AJ            |   |     |     |         |       |        |     |    V2
  |     |       |              |   |     |     |         |       |        |     |    |
  |     |      N3-A            |   |     |     |         |       |        |     |    V1
  |     |       |              |   |     |     |         |       |        |     |    |  \
  |     |      N2J             |   |     |     |         |       |        |     |    VJ  VN-T
  |     |       |              |   |     |     |         |       |        |     |    |    |
  |     |       N2             |   |     |     |         |       |        |     |    |   N3J
  |     |       |              |   |     |     |         |       |        |     |    |    |
  |     |       N1             |   |     |     |         |       |        |     |    |  N3-AJ
  |     |       |              |   |     |     |         |       |        |     |    |    |
  |     |       N*  ---        |   |     |     |         |       |        |     |    |   N3-A
  |     |       |      \       |   |     |     |         |       |        |     |    |    |
  |     |       N*       \     |   |     |     |         |       |        |     |    |   N2J
  |     |       |         |    |   |     |     |         |       |        |     |    |    |
  |     |       |         |    |   |     |     |         |       |        |     |    |    N2 ---
  |     |       |         |    |   |     |     |         |       |        |     |    |    |     \
  |     |       |         |    |   |     |     |         |       |        |     |    |   NLM*   N1
  |     |       |         |    |   |     |     |         |       |        |     |    |    |     |
  |     |       |         |    |   |     |     |         |       |        |     |    |   NLM*   N*
  |     |       |         |    |   |     |     |         |       |        |     |    |    |     |
  |     |       |         |    |   |     |     |         |       |        |     |    |   NLM    A1
  |     |       |         |    |   |     |     |         |       |        |     |    |    |     |
  |     |       |         |    |   |     |     |         |       |        |     |    |   QUAN   |
  |     |       |         |    |   |     |     |         |       |        |     |    |    |     |
 comp   v       n         n    ,   v    art    n         n      cjsb     v    comp   v   quan   n
  To  meet  spectrum specific  , allow  a  30-minute warm-up before attempting to  make any calibrated

            -analyzer  -ations                                                          -measurements
```

Figure 3 Example: a Syntax Tree for a Source Sentence

Note that, a phrase structure grammar with a wide coverage may produce many nodes which simply represent the various bar levels of a special constituents (such as the V2, V1, N2, N1 constructs) or some recursive constructs (such as the N* constructs). Such nodes could be normalized without losing much information in a practical system.

4.2 Syntactic Normalization

Many parse trees that are syntactically identical could be normalized to a normalized syntax tree to reduce the size of the possible parameter space. Such syntactic variants may result from a writing convention, function words, or non-discriminative syntactic information. For instance, some punctuation marks, excessive nodes for identifying the various bar levels in the phrase structure grammar could be deleted or compacted without losing the syntactic relations among constituents. A normalized version of the above syntax tree is shown as follows. We shall call such a normalized syntax tree acquired in the first normalization stage as an NF1 tree.

```
SIMP -----------------------------------------------------------
  |                              \     \                      \
  SB -----------             V     NP ---                  SB ---------------
  |     \       \            |      |     \                |        \        \
  |     V       NP ------    |     DET   NP -------        |        V        VP ----------
  |     |       |       \    |      |     |        \       |        |        |    \       \
  |     |       |        |   |      |     |         |      |        |        |    NLM      NP
  |     |       |        |   |      |     |         |      |        |        |     |        |
 comp   v       n        n   v     art    n         n     cjsb      v        v    quan      n

  To   meet  spectrum  specific allow  a 30-minute  warm-up  before attempting  make  any calibrated
            -analyzer  -ations                                                       -measurements
```

Figure 4 Example: an NF1 Tree (Normalized Syntax Tree)

In the current example, the syntax tree is greatly compacted by retaining only the major syntax structure; a large number of nodes are compacted and re-labelled with representative node labels.

4.3 Semantic Normalization

In semantic analysis, it is desirable that different syntactic structures that are semantically identical could be mapped to the same semantic normal form to further reduce the size of the possible parameter space; discriminative features are then extracted from such normalized constructs for further processing. For instance, it is desirable to normalize the English tense, modal and type information as a feature vector attached to the sentence (or proposition), and normalize the various propositions to a hierarchical feature structure (such as a case frame) to encode the case information, like Agent, Action, Goal, Purpose, and so on. Two sentences with different voices (e.g., active voice vs. passive voice) may also be normalized to a standard form. A normalized semantic tree (referred to as an NF2 tree) of this kind is shown in the following figure.

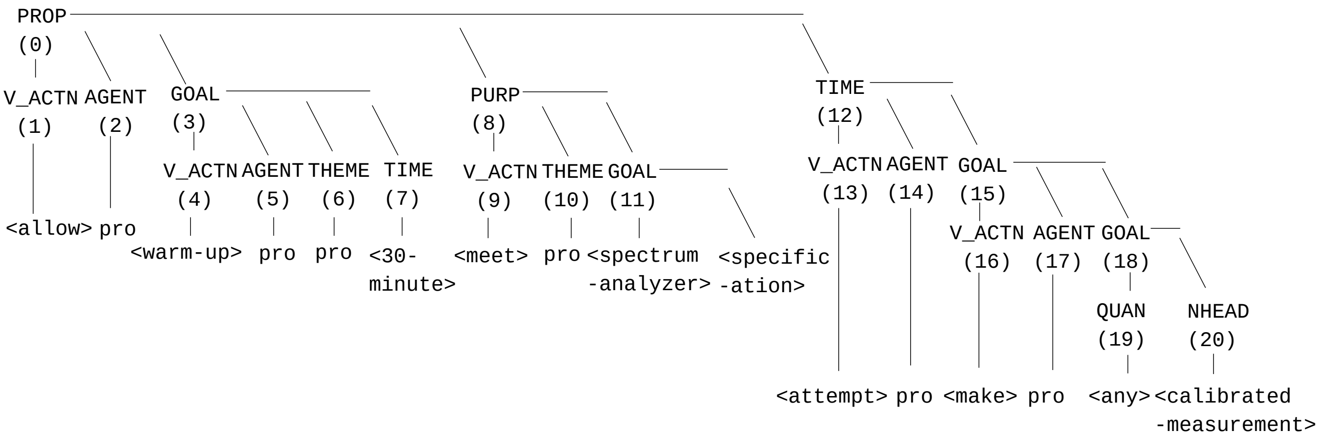

Figure 5 Example: an NF2 Tree (Normalized Semantic Tree)

The PROP (Proposition) frame, for instance, are filled with the Action (V_ACTN) being performed, the Agent (AGENT) who conducted the action, as well as the TIME, GOAL and PURPose for conducting the action. Note that the surface linear order of the nodes is no more retained in such a normal form.

4.4 Target Semantic Tree Selection

Given the normalized semantic tree (NF2 tree) of the source sentence, a proper semantic tree of the target sentence could be selected, among the set of target semantic trees that could be derived from the target grammar under the various discourse context; the selection could be made based on the input source intermediate construct NF2 with some preference score. Optionally, the discourse context and user feedback could be included to select the best mapping among all legal NF2's of the target language. Again, emphasis is placed on the set of legal NF2's that could be derived from the target grammar, instead of a deformed version of the source NF2, to eliminate source dependency. And the process is a 'selection' process rather than a 'derivation' or 'transfer' process from the source semantic tree. By selection, the target NF2 is only selected from legal target NF2's; therefore, the generated target sentence will not be an illegal one. Furthermore, the selection process can also be based on the features derived from the source NF2 tree, rather than the NF2 tree itself as the 'transfer' operation usually does.

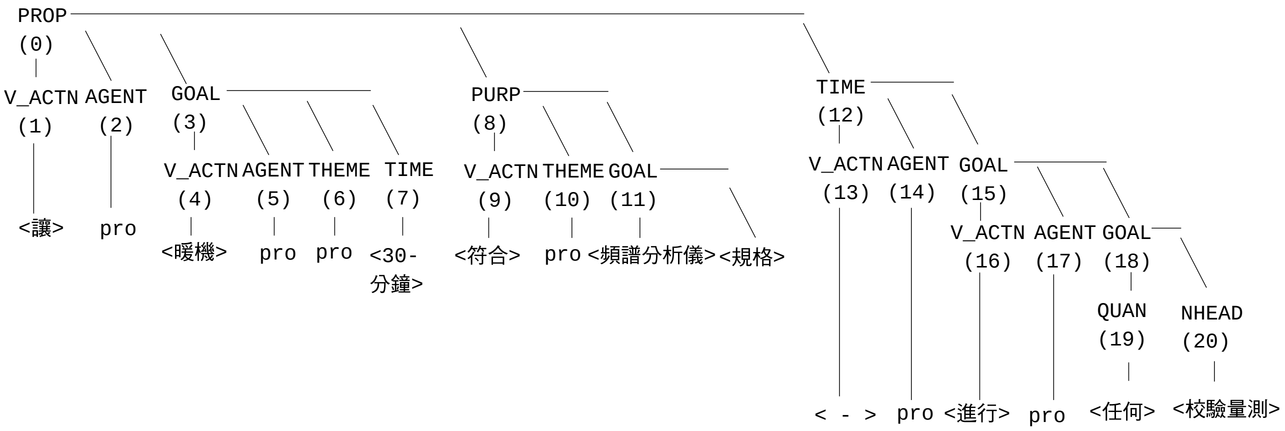

Figure 6 A Preferred Target Semantic Tree Selected from Legal NF2 Trees.

In the above example, the selected target NF2 does not differ much from the source NF2 due to the 2 level normalizations. The major change here is the transfer of word senses in the two languages (where a sense is represented with a pair of angle brackets).

4.5 Normalized Structure Generation

Give a target NF2 that is derivable from the target grammar, the next step would be to choose an appropriate normalized syntax tree for generation. As in the above phase, we could include discourse context and user feedback in this phase to select the tense, mode, voice, type for the sentence, and the most appropriate lexicon for the content words. An NF1 tree of the target sentence generated in this way will contain the major skeletal structure for the target sentence. The following example shows such a skeletal syntax tree.

```
S ---------------------------------------------------------------------------
 |                             \                           \    \      \     \
VADT                         VADT                        VADT    |     NP    VP ---
 |                             |                           |     |     |      |     \
 PP ----                      PP -------------------       |     |     |      |   BAK --
 |      \                      |     \              \      |     |     |      |     |    \
 |      VP ----                |     VP ---          |     |     |     |      |     |     |
 |      |      \               |      |     \        |     |     |     |      |     |     |
 |      |      NP -----        |      |      |       |     |     |     |      |     |     |
 |      |       |       \      |      |      |       |     |     |     |      |     |     |
 |      |      NLM       |     |      |      |       |     |     |     |      |     |     |
 |      |       |        |     |      |      |       |     |     |     |      |     |     |
 pv    vn       n        n    pls    vn      n      loc   advv  vnv    n     vi     q   cltm
 爲了  符合  頻譜分析儀   規格  在    進行    校驗量測    之前   請    讓   儀器   暖機   30   分鐘
```

Figure 7 Normalized Syntax Tree for the Target Sentence

Note again that, the generation step here is actually accomplished by a selection process from a set of legal normalized syntax trees which are derivable from the target grammar. The normalized syntax trees thus generated are still kept within the language defined by the target grammar in a normalized form. Therefore, the final translation output will not be deformed to produce unnatural translation.

4.6 Surface Structure Generation

After the normalized syntax tree is generated, the final syntactic structure could be produced by patching required constituents in a complete syntax tree. The following syntax tree shows one such example. (In this simple case, only two punctuation marks are patched here.)

```
S --------------------------------------------------------------------------------
  |                          \    \                          \   \     \     \     \
 VADT                         |  VADT                         | VADT    |    NP     VP ---
  |                           |    |                          |   |     |     |      |    \
  PP ----                     |   PP -------------------      |   |     |     |      |  BAK --
  |      \                    |    |     \              \     |   |     |     |      |    |    \
  |      VP ----              |    |     VP ----         |    |   |     |     |      |    |     |
  |      |      \             |    |      |     \        |    |   |     |     |      |    |     |
  |      |      NP -----      |    |      |      |       |    |   |     |     |      |    |     |
  |      |       |       \    |    |      |      |       |    |   |     |     |      |    |     |
  |      |      NLM       |   |    |      |      |       |    |   |     |     |      |    |     |
  |      |       |        |   |    |      |      |       |    |   |     |     |      |    |     |
  pv    vn       n        n  cj  pls     vn      n      loc  cj advv   vnv    n    vi     q   cltm
  爲了  符合  頻譜分析儀   規格  ，  在    進行    校驗量測    之前  ，  請    讓   儀器   暖機   30   分鐘
```

Figure 8 Target Syntax Tree

4.7 Morphological Generation

Finally, the terminal tokens of the target syntax tree are de-normalized by including language specific morphological information required in the target language. (In this example, no specific tokens of this kind are inserted, and we have the following output as the preferred translation: 爲了符合頻譜分析儀規格，在進行校驗量測之前，請讓儀器暖機30分鐘。 The Chinese-specific morpheme 「們」, which is used in conjunction with certain nouns to produce their plural form, such as 「同學們」, for example, is generated in this phase.)

4.8 Parameterization Examples Based on the Normalized Constructs

Note that, the above-mentioned normal forms (including the syntax trees) may not be directly used as the atomic units for parameterization. Instead, only some features in the various constructs will be extracted for disambiguation or selection in the processing steps. In syntax tree disambiguation, for instance, it was proposed to decompose a syntax tree into sets of terminal and nonterminal symbols which characterize the snapshots of the various parsing steps, and use the transition probabilities between such sets for evaluating the syntactic scores for the various parses [Su 91a]. In resolving the attachment problem, it was proposed to use a 4-tuple consisting of the features for the main verb, the preposition, and the two noun phrases as the feature vector for disambiguation [Liu 90]. Such parameterization techniques will not be addressed in this paper. Interested readers are referred to some such research [Su 88, Liu 90, Su 91a, Chang 92, 93]. Since we are interested in the training of the translation and transfer knowledge, without loss of generality, we will regard the normal forms as our atomic units throughout this paper.

5. Translation Model

To make it more clear on how such a system removes the source dependency and provides adaptibility to different language styles, the following models show the baseline formulations.

5.1 Optimization Criterion of the Translation Model

Given a source sentence S_i, the Bayesian decision to get the best target sentence T_i would correspond to finding the target sentence that maximizes the conditional probability $P(T_i|S_i)$. To reduce the large number of possible parameters, it is usually essential to normalize syntactically or semantically identical constructs into a normalized form as shown in the above section; and in many cases, it is desirable to decompose such normalized forms into atomic units which only depend on local context [Chang 93]. If we take all the previous normal forms acquired in the translation process into account, the problem is to find the best target sentence T_i that maximizes the following conditional probability :

$$P\left(T_i|S_i, I_1^{i-1}\right)$$

where I_1^{i-1} represents the sets of normal forms $I_1, I_2, \ldots, I_{i-1}$ acquired before for the previous sentence pairs, and

$$I_i = \left\{PT_t(i), NF1_t(i), NF2_t(i), NF2_s(i), NF1_s(i), PT_s(i)\right\}$$

represents one possible combination of the intermediate normal forms for the i^th^ sentence pair, which are derivable from the source and target sentences. The set of the conditional probabilities $P\left(T_i|S_i, I_1^{i-1}\right)$ or probabilities derived from such a form are referred to as the 'parameters' of the systems; their values can be estimated from text corpora. We will show how to estimate such parameters from a bilingual corpus in the next chapter.

When the discourse information, namely I_1^{i-1}, is ignored, we could find the translation T_i that maximizes the following *translation score* [Chang 93]:

$$P\left(T_i|S_i\right) = \sum_{I_i} P\left(T_i, I_i|S_i\right)$$

To simplify the evaluation of the translation score, we can make some reasonable assumptions on the above formula. In particular, if the normalized forms are properly designed, we have:

$$\begin{aligned}
&\sum_{I_i} P\left(T_i, I_i|S_i\right) \\
&= \sum_{I_i} P\left(T_i, PT_t(i), NF1_t(i), NF2_t(i), NF2_s(i), NF1_s(i), PT_s(i)|S_i\right) \\
&= \sum_{I_i} \left\{ P\left(T_i|PT_t(i), NF1_t(i), NF2_t(i), NF2_s(i), NF1_s(i), PT_s(i), S_i\right)\right. \\
&\quad \times P\left(PT_t(i)|NF1_t(i), NF2_t(i), NF2_s(i), NF1_s(i), PT_s(i), S_i\right) \\
&\quad \times P\left(NF1_t(i)|NF2_t(i), NF2_s(i), NF1_s(i), PT_s(i), S_i\right) \\
&\quad \times P\left(NF2_t(i)|NF2_s(i), NF1_s(i), PT_s(i), S_i\right) \\
&\quad \times P\left(NF2_s(i)|NF1_s(i), PT_s(i), S_i\right) \\
&\quad \times P\left(NF1_s(i)|PT_s(i), S_i\right) \\
&\quad \times \left. P\left(PT_s(i)|S_i\right)\right\} \\
&\cong \sum_{I_i} \left\{ \left[P\left(T_i|PT_t(i)\right) \times P\left(PT_t(i)|NF1_t(i)\right) \times P\left(NF1_t(i)|NF2_t(i)\right) \right] \right. && \ldots(1) \\
&\quad \times \left[P\left(NF2_t(i)|NF2_s(i)\right) \right] && \ldots(2) \\
&\quad \times \left. \left[P\left(NF2_s(i)|NF1_s(i)\right) \times P\left(NF1_s(i)|PT_s(i)\right) \times P\left(PT_s(i)|S_i\right) \right] \right\} && \ldots(3)
\end{aligned}$$

where the first three factors in (1) correspond to the generation score, the factor in (2) corresponds to the transfer score and the three factors in (3) correspond to the analysis score for one particular path (i.e., the particular set of intermediate normal forms); the individual factors (from the last factor to the first one), on the other hand, corresponds to the preference to syntactic parsing, syntactic normalization, semantic normalization, semantic tree selection, normalized structure generation, surface structure generation and morphological generation, respectively. In the above derivation, we assumed that, after proper normalization, the intermediate form of the current processing phase could be determined from the information acquired in the last processing step, and the information from other earlier processing steps could be ignored.

In practical situations, we usually choose the T_i whose *maximum path score* (i.e., $\max_{T_i, I_i} P\left(T_i, I_i|S_i\right)$) is highest as the most preferred output, instead of choosing the one whose *sum of path scores* over all possible paths (i.e., $\sum_{I_i} P\left(T_i, I_i|S_i\right)$) is highest. In other word, we would prefer the translation T_i^* of S_i, where

$$
\begin{aligned}
T_i^* &= \underset{\mathrm{T_i}}{\operatorname{argmax}}\left\{ \max_{I_i} \left[P\left(T_i, I_i | S_i \right) \right] \right\} \\
&= \underset{\mathrm{T_i}}{\operatorname{argmax}} \{ \max_{\mathrm{I_i}} \left[P\left(T_i | PT_t(i) \right) \times P\left(PT_t(i) | NF1_t(i) \right) \times P\left(NF1_t(i) | NF2_t(i) \right) \right. \\
&\quad \times P\left(NF2_t(i) | NF2_s(i) \right) \\
&\quad \times \left. P\left(NF2_s(i) | NF1_s(i) \right) \times P\left(NF1_s(i) | PT_s(i) \right) \times P\left(PT_s(i) | S_i \right) \right] \}
\end{aligned}
$$

Note that, if the level-2 normalization of the source and target sentences, namely $NF2_s$ and $NF2_t$, are not a deformed version of their counterparts, then the formulations for bidirectional translation are symmetric. Furthermore, under such circumstances, all the translation output will fall within the range of the sentences that would be produced by native post-editors; therefore, the source dependency is removed.

5.2 Including Discourse Information

We can also take discourse information into account by including the term I_1^{i-1} to Eq. (1)-(3) of the previous section. For instance, if we retain a window size of 2, and we only care to use extra discourse information for scoring the target NF2 tree, then we will, instead, find the best target sentence T_i that maximizes $P\left(T_i | S_i,\ NF2_t(i-2),\ NF2_t(i-1) \right)$. It is easy to derive a similar simplified translation score by following the same procedure in the above section. The major difference is: we will have a factor of $P\left(NF2_t(i) | NF2_s(i), NF2_t(i-2), NF2_t(i-1) \right)$ in the simplified translation score instead of $P\left(NF2_t(i) | NF2_s(i) \right)$ in Eq. (2).

With such a formulation, $NF2_t(i)$ no more depends on $NF2_s(i)$ only. This would be helpful in reducing the source dependency problem further. Through such a mechanism, it is possible to suppress the term dominated by the preference score $P(NF2_t(i)| NF2_s(i))$ with the target discourse context. Of course, one could use the features extracted from $NF2_s(i)$, $NF2_t(i-2)$, and $NF2_t(i-1)$, instead of the normal form themselves, to select the preferred $NF2_t(i)$. Using discourse information other than $NF2_t(i-2)$ and $NF2_t(i-1)$, such as $NF1_t(i-1)$, in deciding the preference of other intermediate forms is also possible in the above derivation.

6. Two-Way Training

To implement a corpus-based statistics-oriented parameterized MT system as characterized above with a two-way training method would encounter the following problems:

(1) how to parameterize the normalized structures,

(2) how to estimate the parameters, and

(3) how to acquire the tagged corpora

The estimation based on a complete syntax tree or normal form tree is unlikely to be practical in a real system. Therefore, the various factors in the previous optimization function must be simplified to make the evaluation feasible. Several general techniques are useful in this regard. In particular, one could decompose such hierarchical structures into atomic units ([Su 91a, Chang 93]), simplify the conditional probabilities by taking a finite window around the current atomic unit under consideration, the system can then be parameterized.

Parameters described above can be estimated from tagged corpora with an MLE (Maximum Likelihood Estimator). Additionally, the baseline model could be refined with adaptive learning methods, which adjust the parameter sets according to misclassified instances to reduce the error rate and increase system robustness ([Amari 67, Juang 92]). Interested readers are referred to related research as in [Chiang 92, Su 91b, 94a]. Since every steps of the above architecture may be performed based on various assumptions, there remains a large number of research issues to be exploited. Without loss of generality, we shall consider the intermediate representations like parse trees and normal forms as atomic units, and propose a model for training the translation parameters with a bilingual corpus in the following sections.

6.1 Two-End Constraint Optimization as a Solution to Two-Way Design

Formally, the estimation (or training) problem can be formulated as the process for finding the set of parameters that maximize an optimization function for all translation pairs. In our case for an MT, this corresponds to finding a parameter set Λ_{MAX} that maximizes the following likelihood function over all translation pairs:

$$\begin{aligned}\Lambda_{MAX} &= \operatorname*{argmax}_{\Lambda} P\left(T_1^N \middle| S_1^N, \Lambda\right) \\ &= \operatorname*{argmax}_{\Lambda} \sum_{I_1^N} P\left(T_1^N, I_1^N \middle| S_1^N, \Lambda\right)\end{aligned}$$

where S_1^N refers to the set of N source sentences, $S_1, S_2, ..., S_N$, T_1^N refers to the target sentences $T_1, T_2, ... T_N$, and I_1^N stands for one possible set of intermediate representations $I_1, ..., I_N$ for aligned bilingual corpus $\left(S_1^N, T_1^N\right)$. To simplify the estimation, it is reasonable to consider only the most preferred intermediate forms for each sentence pair derived from the source sentence and the preferred target sentence (known as a Viterbi training method [Rabiner 86]). In other words, we could estimate the parameters to maximize the following likelihood function:

$$\Lambda_{MAX} = \operatorname*{argmax}_{\Lambda} \left[\max_{I_1^N} P\left(T_1^N, I_1^N \middle| S_1^N, \Lambda\right) \right].$$

The likelihood function can be further simplified as follows to make it feasible.

$$
\begin{aligned}
&P\left(T_1^N, I_1^N \,\middle|\, S_1^N, \Lambda\right) \\
&= P\left(T_1^N | I_1^N, S_1^N, \Lambda\right) \cdot P\left(I_1^N | S_1^N, \Lambda\right) \\
&= \prod_{i=1}^{N} \left[P\left(T_i | T_1^{i-1}, I_1^N, S_1^N, \Lambda\right) \cdot P\left(I_i | I_1^{i-1}, S_1^N, \Lambda\right) \right] \\
&\cong \prod_{i=1}^{N} \left[P\left(T_i | I_{i-m}^{i}, \Lambda\right) \cdot P\left(I_i | I_{i-m}^{i-1}, S_i, \Lambda\right) \right],
\end{aligned}
$$

where we decompose the likelihood function into two factors, one corresponding to the probability of the output translation given the internal normal forms, and the other corresponding to the probability of intermediate form I_i given the intermediate forms of the previous sentences and the source sentence S_i. If we ignore the discourse context (i.e., m=0), we have the following likelihood function:

$$
\begin{aligned}
&P\left(T_1^N, I_1^N \,\middle|\, S_1^N, \Lambda\right) \\
&\cong \prod_{i=1}^{N} \left[P\left(T_i | I_i, \Lambda\right) \cdot P\left(I_i | S_i, \Lambda\right) \right]
\end{aligned}
$$

Under such circumstances, we can easily derive the following simplified likelihood function with a similar method as in section 5:

$$
\begin{aligned}
&\prod_{i=1}^{N} \left[P\left(T_i | I_i, \Lambda\right) \cdot P\left(I_i | S_i, \Lambda\right) \right] \\
&\cong \prod_{i=1}^{N} \Big\{ \left[P\left(T_i |\; PT_t(i), \Lambda\right) \times P\left(PT_t(i) | NF1_t(i), \Lambda\right) \times P\left(NF1_t(i) | NF2_t(i), \Lambda\right) \right] \\
&\quad \times \left[P\left(NF2_t(i) | NF2_s(i), \Lambda\right) \right] \\
&\quad \times P\left(NF2_s(i) | NF1_s(i), \Lambda\right) \times P\left(NF1_s(i) | PT_s(i), \Lambda\right) \times P\left(PT_s(i) | S_i, \Lambda\right) \Big] \Big\}
\end{aligned}
$$

If the discourse information is considered with a window size of two, we then have the following likelihood function for including such extra information:

$$
\begin{aligned}
&P\left(T_1^N,\ I_1^N \,\middle|\, S_1^N,\ \Lambda\right) \\
&\cong \prod_{i=1}^{N} \left[P\left(T_i | I_{i-2}^{i},\ \Lambda\right) \cdot P\left(I_i | I_{i-2}^{i-1},\ S_i,\ \Lambda\right) \right]
\end{aligned}
$$

and the parameters could be adjusted to maximize the likelihood function which includes a factor like

$$
P\left(NF2_t(i) | NF2_s(i), NF2_t(i-2), NF2_t(i-1), \Lambda\right)
$$

that reflects the discourse information for $NF2_t$ (or optionally $NF1_t$ and $NF2_s$, and so on) in the previous derivation.

If we have the corpora annotated with the most preferred I_i for each sentence pair, the estimation of the parameters would be much easier: we could simply count the number of occurrences of the various intermediate forms; the conditional probabilities in Λ_{MAX} are just the relative ratio between different counts.

A practical problem is: such annotated corpora may not exist due to the high construction cost, or the amount of annotated corpora is limited, which may induce large estimation errors. Since an un-annotated bilingual corpus is much easier to set up than a fully annotated corpus, one promising solution to such a problem is to train the parameters only with a bare bilingual corpus, and consider such an optimization procedure as a two-end constraint optimization problem. By two-end constraint, we mean that we are given only a parallel corpus of the source sentence (S_i) and the preferred target translation (T_i), including the preferred lexicon information, the functional forms of the syntax structures (defined by a phrase structure grammar) and the functional forms of the semantic trees. Here, we have only the two-end constraints on the given (S,T) pairs; the other intermediate representations are left unspecified. We want to find the best parameter set that maximizes the likelihood for the preferred target sentences to be generated by the source sentences.

6.2 A Viterbi Training Approach

In the case that the bilingual corpus is un-annotated, the parameters can be acquired by unsupervised learning methods, like the EM algorithm [Dempster 77]. The general steps of an EM algorithm are to enumerate all possible alternative analyses for each sentence pair, then estimate the expected number of occurrence of different constructs based on all the alternative analyses. The parameters that maximize the optimization function are specified by a function of those expected numbers of occurrences, and this set of parameters is regarded as the optimal parameter set in the current iteration; it is then used to evaluate the path scores for the next iteration. Such process is repeated until the parameters converge. Because of the large number of possible combinations of the intermediate forms, a Viterbi training approach [Rabiner 86] is preferred, instead of the EM approach, for the two-end parameter training problem:

1. Initial Selection: For each (S_i, T_i) pair, we derive all possible parse trees, NF1 trees and NF2 trees for S_i and T_i, respectively. And randomly choose a path along the derivation process from the sentences toward the NF2 trees. The source and target NF2 trees randomly selected in this way are considered a transfer pair for the translation.
2. Estimation: The parameters, estimated with the Maximum Likelihood Estimator (MLE) for the system, are estimated from the corresponding transfer pairs, parse trees and normal forms along the selected translation path. The parameters are uniquely determined once the translation paths are specified.
3. Re-selection: Compute the translation scores for the various possible paths, which specify the combination of parse tree, NF1 and NF2, with the new parameters acquired in step 2, and select the path with the largest translation score.
4. Re-estimation: go to step 2, unless the parameters converge to a particular stopping criterion.

Several variants could be adopted to make the above procedure better. For instance, the Initial Selection step could start with a small annotated bilingual corpus as the seed [Su 94b]. This annotated seed corpus is then mixed with the other untagged corpus for estimation. In addition, to eliminate the large search space in the above process, some of the low-score combinations could be eliminated from consideration as the process proceeds; the number of truncated candidates could be a time increasing function as the parameters are estimated better and better.

6.3 Feedback Controlled System with Adaptive Training

Although the above estimation procedure maximizes the likelihood for the source sentences in the training corpus to generate the target sentences, it does not mean that they are most preferred by all users under all context. In fact, different customers may have their own preferred styles of translation; therefore, different users will measure the translation quality differently, based on their own preference. This means that we need a way to adapt the parameters to minimize a user-specified 'distance' measure between the translation output of the system with a user preferred translation. In other words, the user feedback should be included to the system to adjust the system parameters so that the user style could be respected in the translation output. The following figure shows such a system for incorporating user feedback.

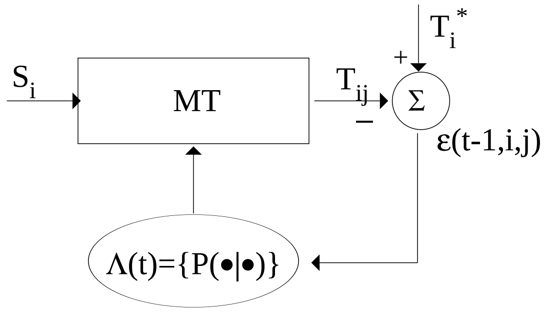

Figure 9 Feedback Controlled Parameterized MT Architecture

The differences between the preferred output translation T_i* of a particular user for an input sentence S_i and its corresponding outputs T_{ij} produced by the MT system is taken into consideration. The difference or error (ε) between the two translations is then used to define a distance measure between them; a cost associated with such a distance can be defined, and an amount of adjustment to the parameters involved in the score evaluation could be computed; the various parameters related to the score evaluation are then adjusted accordingly. Such a process will repeat until the preferred translations have the highest score so that the distances of the preferred translation and the best translation produced by the MT system could be minimized. (The time indices *t* and *t*-1 reflect such an iterative nature of the training procedure). Alternatively, the parameters could be further adjusted until the scores of the preferred translations exceed their competitors by a prescribed margin to enhance the robustness of the system [Su 91b, 94a].

Such an adaptive learning method, generally characterized as a probabilistic descent method [Amari 67, Juang 92, Su 91b, 94a], had been applied to some natural language processing applications [Chiang 92, 94a, 94b, 95, Lin 92, Su 91b, 94a]. It is observed to have increased the discrimination power and robustness of the system. In the current proposal, it is adopted to incorporate the individual user style into the parameter sets trained from the previous Viterbi training procedure.

The distance measure and the cost function could be defined differently, according to user preference. For instance, one can use the differences of the translation scores, as defined in section 5, between the preferred output translation and the translations produced by the MT system as the distance measure, and define the cost to be nearly one when the distance is large and a cost to be nearly zero when the distance is small [Amari 67]. In this way, we could minimize the global sentence error rate of the system as measured by a particular user or customer. Alternatively, one could also use other distance measures, for instance, the number of editing operations required to correct the translation output to the preferred style. With such a distance measure, the number of editing strokes could be minimized [Su 92b]. Therefore, such a distance measure is particularly useful for handling user feedback from on-line post-editors.

In real applications, there are two major types of users, whose feedback need to be handled differently. The first kind of users are the customers who use an MT system for translating a large volumes of text material and whose complaints about the translation output are feedback in batch. The other kind of users are those post-editors, whose complaints need to be resolved on-line. We therefore have two different modes to include user feedback into the system. In the first case, the user feedback should be gathered in batch and the parameters are adjusted with all the problematic translations jointly considered. Such a batch mode operation is called *training by epoch* [Schalkoff 92]. In the latter case, the parameters are changed as each new user feedback or complaint is given. Such an on-line mode of operation is referred to as *training by sample* [Schalkoff 92].

With the above training methods, it is possible to resolve the two-end constraint optimization problem to estimate the underlying parameter sets without much human effors in constructing the annotated corpora. Besides, a feedback controlled parameterized MT system, as proposed here, is feasible for producing translation output that will not be strongly influenced by the source language; particular user-specific requirements could also be properly included without changing the system modules for such adaptation.

7. Concluding Remarks

In this paper, we propose a feedback controlled parameterized MT system that respects the native styles of the target language as well as user specified styles. A new architecture is proposed for implementing such a system. The training method for estimating the parameters of the system, using a corpus-based statistics-oriented (CBSO) two-way design philosophy, is also exploited to acquire the underlying translation and transfer knowledge from a bilingual corpus. With such a paradigm, many traditional transfer-based MT modules could be parameterized to gain more flexibility and better performance. With the superiority in knowledge acquisition, adaptability, optimality and bidirectionality, this new architecture is expected to play an important role in designing the MT systems of the next generation. Currently, we are making the BehaviorTran MT system [Su 90, Chen 91, Chang 93], which was used for translation services for international computer companies, evolve toward such a new architecture.

References

[Amari 67] Amari, S. "A theory of adaptive pattern classifiers", *IEEE Trans. on Electronic Computers*, vol. EC-16, no. 3, pp. 299-307, June 1967.

[Brown 90] Brown, P., J. Cocke, S. D. Pietra, V. J. D. Pietra, F. Jelinek, J. D. Lafferty, R. L. Mercer, and P. S. Roossin, "A Statistical Approach to Machine Translation", *Computational Linguistics*, vol. 16, no. 2, pp.79-85, June 1990.

[Chang 92] Chang, J.-S., Y.-F. Luo and K.-Y. Su, "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, 1992.

[Chang 93] Chang, J.-S. and K.-Y. Su, "A Corpus-Based Statistics-Oriented Transfer and Generation Model for Machine Translation," *Proceedings of TMI-93*, pp. 3--14, 5th Int. Conf. on Theoretical and Methodological Issues in Machine Translation, Kyoto, Japan, July 14--16, 1993.

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

[Chiang 92] Chiang, T.-H., Y.-C. Lin and K.-Y. Su, "Syntactic Ambiguity Resolution Using A Discrimination and Robustness Oriented Adaptive Learning Algorithm", *Proceedings of COLING-92*, vol. I, pp. 352--358, 14th Int. Conference on Computational Linguistics, Nantes, France, 1992.

[Chiang 94a] Chiang, Tung-Hui, Yi-Chung Lin, and Keh-Yih Su, "On Jointly Learning the Parameters in a Character-Synchronous Integrated Speech and Language Model," *Proceedings of 1994 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp.(II) 29-32, Adelaide, Australia, 19-22 April 1994.

[Chiang 94b] Chiang, Tung-Hui , Yi-Chung Lin, and Keh-Yih Su, "A Study of Applying Adaptive Learning to a Multi-module System," *Proceedings of 1994 International Conference on Spoken Language Processing*, pp. 463-466, Yokohama, Japan, Sep. 18-22, 1994.

[Chiang 95] Chiang, T.-H., Y.-C. Lin and K.-Y. Su, "Robust Learning, Smoothing and Parameter Tying on Syntactic Ambiguity Resolution," to appear in *Journal of ACL*, 1995.

[Dempster 77] Dempster, A. P., N. M. Laird and D. R. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm," *Journal of the Royal Statistical Society*, 39 (B), pp. 1-38, 1977.

[Hsu 94] Hsu, Yu-Ling Una and Ming-Wei Guo, "New Architecture of BehaviorTran," *BDC Technical Report: LG-94-01-NF-01*, Behavior Design Corporation, Hsinchu, Taiwan, ROC.

[Hutchins 86] W.J. Hutchins, *Machine Translation: Past, Present, Future*, Ellis Horwood Limited, West Sussex, England, 1986.

[Juang 92] Juang, B. H. and S. Katagiri, "Discriminative Learning for Minimum Error Classification," *IEEE Trans. on Acoustics, Speech and Signal Processing*, vol. 40, no. 12, pp. 3043-3054, Dec. 1992.

[Lin 92] Lin, Yi-Chung, Tung-Hui Chiang and Keh-Yih Su, "Discrimination Oriented Probabilistic Tagging," *Proceedings of ROCLING-V*, ROC Computational Linguistics Conference V, pp. 87--96, 1992.

[Liu 90] Liu, C.-L., J.-S. Chang and K.-Y. Su, "The Semantic Score Approach to the Disambiguation of PP Attachment Problem," *Proceedings of ROCLING-III*, pp. 253-270, Taipei, R.O.C., 1990.

[Rabiner 86] Rabiner, L. R., J. G. Wilpon and N.-H. Juang, "A Segmental k-Means Training Procedure for Connected Word Recognition," AT&T Tech. Journal, vol. 65, no. 3, pp. 21-31, May-June 1986.

[Schalkoff 92] Schalkoff, Robert J., *Pattern recognition: statistical, structural and neural approaches*, John Wiley & Sons, 1992.

[Somers 93] Somers, Harold, "Current Research in Machine Translation", *Machine Translation*, vol. 7, pp. 231-247, Kluwer Academic Publishers, 1993.

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

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

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

[Su 91b] Su, K.-Y., and C.-H. Lee, "Robustness and Discrimination Oriented Speech Recognition Using Weighted HMM and Subspace Projection Approach," *Proceedings of IEEE ICASSP-91*, vol. 1, pp. 541-544, Toronto, Ontario, Canada. May 14-17, 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, 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, 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.

[Su 94a] Su, K.-Y. and C.-H. Lee, "Speech Recognition Using Weighted HMM and Subspace Projection Approaches," *IEEE Transactions on Speech and Audio Processing*, pp. 69--79, vol. 2, no. 1, part 1, Jan. 1994.

[Su 94b] Su, K.-Y., T.-H. Chiang and J.-S. Chang, "Introduction to Corpus-based Statistics-oriented (CBSO) Techniques," Pre-Conference Workshop on Corpus-based NLP, ROC Computational Linguistics Conference VII, National Tsing-Hua Univ., Taiwan, ROC., Aug. 1994.