[*MT Summit III*, Washington, DC, USA, July 1991]

ArchTran: A Corpus-based Statistics-oriented English-Chinese Machine Translation System

Shu-Chuan Chen*, Jing-Shin Chang, Jong-Nae Wang*, and Keh-Yih Su****

*Behavior Design Corporation
2F, 28 R&D Road II
Science-based Industrial Park
Hsinchu, Taiwan 300, R.O.C.

**Department of Electrical Engineering
National Tsing Hua University
Hsinchu, Taiwan 300, R.O.C.
kysu@ee.nthu.edu.tw

Abstract

The ArchTran English-Chinese Machine Translation System is among the first commercialized English-Chinese machine translation systems in the world. A prototype system was released in 1989 and currently serves as the kernel of a value-added network-based translation service. The main design features of the ArchTran system are the adoption of a mixed (bottom-up parsing with top-down filtering) parsing strategy, a scored parsing mechanism, and the corpus-based, statistics-oriented paradigm for linguistic knowledge acquisition. Under this framework, research directions are toward designing systematic and automatic methods for acquiring language model parameters, and toward using preference measure with uniform probabilistic score function for ambiguity resolution. In this paper, the underlying probabilistic models of the ArchTran designing philosophy will be presented.

1 Introduction

The ArchTran Machine Translation System is the first of its kind research launched in Taiwan, and it is among the first commercialized English-to-Chinese systems in the world.

The research on ArchTran began as a joint effort between National Tsing Hua University, Taiwan, and Behavior Tech Computer Corporation (BTC) in May, 1985. Research was later transferred to Behavior Design Corporation (BTC R&D Center) founded in February of 1988 to continue improvements on the system.

As a research conducted in a private organization, the goal of the system is a commercially viable system. After four years of research, a prototype system was released in 1989 and serves as the kernel of a value-added network (VAN)-based translation service. Currently, ArchTran has established a customer base of several internationally renowned computer companies and with several others at testing stage. While the primary domain for ArchTran is computer manuals and related documents, services will be expanded to other technical fields in the near future.

The system is running on the Sun Workstations and written in C language for its portability. The raw translation is post-edited on PCs, which are connected to local or public network to form the VAN-based translation workstations.

The research of the ArchTran system has the characteristics of being corpus-based and statistics-oriented. Under this framework, research directions are toward designing systematic and automatic approaches for acquiring language model parameters, and toward using preference measure with uniform probabilistic score function for ambiguity resolution.

2 System Description

Some of the distinctive features of this transfer-based, batch-processing MT system are the adoption of a mixed bottom-up parsing with top-down filtering parsing strategy, a scored parsing mechanism and the corpus-based, statistics-oriented paradigm for linguistic knowledge acquisition [Su *et al.*, 1990a, Su and Chang, 1990b].

The most notable design features in ArchTran will be briefly explained. They include strategies concerning the following aspects of the system (1) grammar (2) lexicon (3) parsing (4) probabilistic language models.

2.1 Grammar

The analysis grammar of ArchTran is an ATN-style augmented context-free phrase structure grammar [Hsu and Su, 1986]. One of its main features is *grammar splitting* [Su and Chang, 1990b], which divides the analysis grammar into a number of subgrammars.

Dividing grammar rules into several independent groups according to their syntactic categories is essential, especially if we want to parse a *nonsentential constituent* such as an NP or VP. In the ArchTran system, the subgrammars are activated by a special action,

called "wake-up," in the subparsing tables of a revised LR parser. The grammar is divided into subgrammars for constructing major constituents like S, NP, and VP. Major constituents are basically constructed bottom-up, but top-down predictions are made for activating other major constituents.

Hence, grammar splitting provides a convenient way to construct each major constituent independently. It also eases the implementation of the mixed parsing strategy, bottom-up parsing with top-down filtering.

2.2 Lexicon

The main features of ArchTran's lexicon are *hierarchical* and *unification-based* [Chen *et al.*, 1989]. ArchTran is used to translate an ample variety of texts. The diversity in text types and topics, among other things, poses problem on handling different senses, usages, and customer-specified translations of a given lexical item.

To cope with this problem, ArchTran designs a unification-based method in which dictionaries are divided into four categories and are organized in a hierarchy of decreasing specificity. They are project dictionary, customer dictionary, technical dictionary, and general dictionary. A given lexical item and its various attributes are stored in the proper dictionaries according to their domain of use. Unification is then employed to select the appropriate attributes from the appropriate lexicons during translation.

The way in which the system dictionaries are constructed and the way in which the dictionary information is unified are useful in dealing with the problem of word sense ambiguity and customized translation.

2.3 Parsing

The parsing strategies in ArchTran are featured by partial parsing, scored parsing, and sequential truncation.

2.3.1 Partial Parsing

Grammar splitting makes partial parsing possible because each constituent can be parsed independently. This feature is important especially in dealing with real-life texts, which may contain nonsentential fragments. Since the processing unit is no longer confined to a sentence, we can perform many special treatments more flexibly. For example, we can verify whether an NP exists in a split idiom such as "turn NP on" if the NP subgrammar can be activated independently. The treatment of nonsentential input (e.g., titles and parenthetical text) also benefits from such a feature. When combined with the chart mechanism which retains the partial parses, it is also possible to realize the fail-soft strategy for error recovery. Such a strategy is important for occasional ungrammatical yet acceptable input sentences to the ArchTran system.

2.3.2 Scored Parsing and Sequential Truncation

In ArchTran, a scored truncation strategy [Su *et al.*, 1989, Su *et al.*, 1990a] is employed to reduce the number of invalid ambiguities by searching only the subspace where the best analysis is most likely to be found. The scored truncation strategy traverses only one path at a time. The searching process is directed by the score function and a set of dynamically adjustable thresholds. These thresholds serve to truncate unlikely parses by comparing the threshold of each stage and the current accumulative partial score at the end of each step. After the first parse is obtained, each new parse is also compared with the analysis which has the highest overall score. The best analysis will be returned at the end of the parsing process. Such strategy is called "sequential truncation". With such techniques, only a small fraction of the most promising parsing state space is traversed.

The scored truncation strategy can provide significant improvement on parsing efficiency. However, a good scoring mechanism for preference assignment is essential. In fact, the most critical linguistic problem in machine translation is to find the most appropriate interpretation from hundreds or thousands of ambiguous constructions.

Besides using linguistic knowledge for ambiguity resolution, ArchTran adopts a probabilistic preference measure called "score function" [Su and Chang, 1988, Su *et al.*, 1991b, Liu, 1989, Liu *et al.*, 1990, Chang, 1990] to provide an objective measure for ambiguity resolution. This function integrates lexical, syntactic, and semantic knowledge with a uniform formulation to assign a preference measure to each interpretation. The formulation and designing philosophy are outlined in the following sections.

3 Probabilistic Translation Model

ArchTran is based on conventional transfer-based MT systems except for the above mentioned features; many rules are encoded in the system to take care of the various linguistic problems in the system. However, as the system scales up, the rule-based approach suffers from some problems. In particular,

- It is hard to deal with uncertainty knowledge due to the lack of objective preference measure.
- It is hard to deal with complex and irregular knowledge. Exceptions to the knowledge occur from time to time.
- It is hard to maintain consistency of the large amount of fine-grained knowledge among different persons at different times.
- It is hard and costly to acquire the large amount of fine-grained knowledge with human intervention.
- There is no systematic way to acquire linguistic knowledge as proposed in various formalisms.

Hence, the problems of ambiguity resolution, ill-formedness and error recovery become harder and harder to tackle. The knowledge acquisition problem turns out to be the major bottleneck for scaling up the MT system. To resolve such problems, the ArchTran designing philosophy is redirected toward a corpus-based statistics-oriented (CBSO) approach. The probabilistic approach is adopted for a number of reasons:

- Objective preference measure for scored parsing is easier to establish.

- Consistency can be easily acquired even in large-scale system by providing suitable probabilistic language models.
- Automatic or semi-automatic training processes are possible. Hence, the burden and cost for humans to acquire the fine-grained knowledge can be reduced.
- Well-established statistical theories and methodologies are available. Hence, systematic approaches for improving the system are possible.

In sum, we think that humans are competent in general language modeling while computers are effective in processing massive data. Therefore, it is appropriate to take advantages of well-recognized linguistic phenomena, setup probabilistic language models by humans, and estimate the parameters of the probabilistic models from large corpora.

The general problem of machine translation is to find the best mapping between a language pair under the known language characteristics of the source and target languages and a mechanism for defining appropriate mapping. Hence, the probabilistic view of the translation problem in ArchTran formulation is to find the target sentence that maximizes the following *translation score*:

$$
\begin{aligned}
& P \quad (Cw_1^{n_C} | Ew_1^{n_E}, ELM, TM, CLM) \\
& = \sum\sum P(Cw_1^{n_C}, I_C, I_E | Ew_1^{n_E}, ELM, TM, CLM) \\
& = \sum\sum P(Cw_1^{n_C} | I_C, CLM, I_E, Ew_1^{n_E}, ELM, TM) \\
& \quad \times P(I_C | I_E, TM, Ew_1^{n_E}, ELM, CLM) \\
& \quad \times P(I_E | Ew_1^{n_E}, ELM, TM, CLM) \\
& \approx \sum\sum P(Cw_1^{n_C} | I_C, CLM) \qquad (generation) \\
& \quad \times P(I_C | I_E, TM) \qquad (transfer) \\
& \quad \times P(I_E | Ew_1^{n_E}, ELM) \qquad (analysis)
\end{aligned}
$$

In the above formulation, $Ew_1^{n_E}$ and $Cw_1^{n_C}$ are the source (English) and target (Chinese) sentences of length n_E and n_C; ELM and CLM are the source and target language models; I_E and I_C are the intermediate representations (or *interpretations*) of the source and target languages, respectively; and TM is the transfer model from source language to target language.

It will be difficult to evaluate the overall translation score with all factors jointly considered. Fortunately, by introducing the intermediate representations of the source and target (i.e., I_E and I_C), the translation problem can be resolved through three phases. In each phase, the best candidate or the top-N candidates of the preceding phases can be used to reduce the searching efforts of the best solution. This is formulated as the three terms in the above formula. The decomposed terms in the formula can be regarded as the *analysis score* ($P(I_E|Ew_1^{n_E}, ELM)$), *transfer score* ($P(I_C|I_E, TM)$) and *generation score* ($P(Cw_1^{n_C}|I_C, CLM)$), respectively. Hence, the above formulation provides a probabilistic ground for formulating the translation problem and the transfer-based approach in probabilistic domain. The Bayesian view implied in the above formulation also guarantees the optimality of the formulation in statistic sense.

To apply the formulation, the main task is to find effective encoding schemes for the various variables involved in the formula, and to use various techniques to estimate the probabilities in the formula. The linguistic part requires that we define nonredundant and discriminant features for encoding the "semantics" of the sentences, extract abstractness of the language in terms of probabilistic terminologies and develop probabilistic language models that well characterize the linguistic problems. In computational part, the main task is to develop automatic approaches for preparing annotated corpora, estimate the parameters of the language models, sometimes from sparse data, and adaptively adjust the parameters so that they are discriminant and robust enough even for input text from unseen domain.

These problems can be handled in quite a standard way. The general approach of ArchTran in dealing with these problems is to:

- Develop abstractness for the linguistic problems to be handled.
- Decompose the abstract object into statistically measurable events.
- Evaluate the transition probability between the events.
- Adaptively adjust the estimated parameters to acquire robust parameter sets and hence improve the performance of the system in unseen domain.

Because the procedures can be established systematically, it is easy to compile the large volume of linguistic knowledge through the standard procedures once the linguistic models are established. The *score function* paradigm [Su and Chang, 1988, Su *et al.*, 1989, Su *et al.*, 1990a], which is a major mechanism for resolving general ambiguity problem in ArchTran, serves well as a good example for showing such designing philosophy.

3.1 Score Function

The analysis phase is probably the most difficult one in machine translation system. The lack of objective measure and systematic approach to integrate various knowledge sources poses practical problems in applying the various linguistic formalisms. ArchTran tackles these problems by introducing a *score function*. It can be regarded as a possible realization of the "analysis score" mentioned in the previous section.

Intuitively, if we can encode a particular interpretation with a set of lexical, syntactic and semantic primitives, then the main task in analysis is to find the most probable interpretation for the input strings. Hence, we formulate the analysis problem as finding an interpretation that maximizes the following *score function*:

$$
\begin{aligned}
Score \equiv \; & P(Sem, Syn, Lex | Words) \\
\equiv \; & P(Sem | Syn, Lex, Words) \qquad (semantic\ score) \\
\times \; & P(Syn | Lex, Words) \qquad (syntactic\ score) \\
\times \; & P(Lex | Words) \qquad (lexical\ score) \\
\equiv \; & S_{sem} \times S_{syn} \times S_{lex} \\
\approx \; & P(Sem|Syn) \times P(Syn|Lex) \times P(Lex|Words)
\end{aligned}
$$

where *Lex*, *Syn*, and *Sem* are the sets of lexical, syntactic, and semantic primitives used to encode the semantic interpretation of a sentence (*Words*). The score

function takes advantage of all lexical, syntactic and semantic knowledge sources, instead of using individual heuristic preference measures or probabilistic measures for lexical, syntactic, and semantic components. To use this formulation, appropriate encoding schemes must be selected and computational techniques must be used to estimate the probabilities. These tasks are handled in ArchTran as follows.

3.1.1 Lexical Score

The basic lexical feature for analysis is the lexical categories (or parts of speech) of the input tokens. Hence, it is intuitive to formulate the lexical score as:

$$\begin{aligned} S_{lex} &\equiv P(Lex|Words) \equiv P(c_1^n|w_1^n) \\ &= \prod_i P(c_i|c_1^{i-1}, w_1^n) \end{aligned} \quad (1)$$

(c_1^n stands for the sequence of lexical categories $c_1 \cdots c_n$)

Lexical disambiguation with such formulation has been explored with great success [Garside *et al.*, 1987, DeRose and Steven, 1988, Church, 1988]. In these works, the general approach is to regard the general term $P(c_i|c_1^{i-1}, w_1^n)$ as approximately equal to $P(c_i|c_{i-s}^{i-1}) \times P(c_i|w_i)$ where s is the scope or window size of the preceding neighbors of the current lexical category c_i. The first term is called *contextual probability* and the last term is called *lexical probability* [Church, 1988]. In order to reduce searching effort, dynamic programming technique is used to search the network expanded by the various alternative parts of speech. The most appropriate parts of speech are then acquired from the best "path" corresponding to the highest lexical score.

Instead of regarding the lexical probability and contextual probability as two components of the lexical score, we consider these two types of probabilities as two different types of estimation to the original lexical score. These estimations should be smoothed in a transformed domain so as to obtain a better estimation of the lexical score. The following generalized nonlinear smoothing form is proposed in ArchTran formulation based on such philosophy:

$$g(P(c_i|c_1^{i-1}, w_1^n)) \equiv \lambda g(P(c_i|w_i)) + (1-\lambda) g(P(c_i|c_{i-s}^{i-1}))$$

where λ is the *lexical weight* to the lexical probability database, $1-\lambda$ is the *contextual weight*, and g is a transform function. Currently, the transform function is a log function.

The reason for adopting the smoothed form is purely computational:

- It requires several strong independency assumptions to approximate the general term of the lexical score with the product of the lexical probability and contextual probability. This requirement may not be easy to be satisfied.
- The dynamic ranges of the two lexical databases may vary drastically. Hence, they need compression or expansion into a transformed domain.
- The generalized form reduces to the formulation of [Garside *et al.*, 1987, Church, 1988] when the transform function is the log function and λ is 1/2.

By regarding the formulation as a smoothing problem, we find that the best results are acquired when λ is 0.6 for Brown Corpus, and 0.7 for our own corpus [Su *et al.*, 1991b], rather than $\lambda = 1/2$. Thus, it allows us to acquire better performance while using the same amount of probabilistic knowledge.

This example shows how computational techniques in statistics can help improving the performance of the formulation.

3.1.2 Syntactic Score

It is harder to extract the abstractness of "syntax" and formulate the syntactic score. The simpliest way to evaluate the syntactic score is to use a stochastic context-free grammar: Each production rule is associated with a probability, and the product of the rule probabilities of a syntax tree is regarded as its syntactic score [Fu, 1982]. Such formulation suffers from two serious problems in dealing with natural languages.

First, the contextual information, which is very important for natural language, is not encoded in the formulation. This might not be a practical strategy as far as natural language is concerned.

Second, the correlation between successively applied production rules is not considered. In general, even if a language can be represented with a context-free grammar, it does not mean that the rules are used in context-free (statistically independent) manner. A stochastic context-free grammar implicitly assumes that the use of each rule is *independent* of the use of the preceding rules, and thus the overall syntactic score is evaluated as the product of the rule probabilities. This results in a *normalization* problem. In other words, if the independency assumption is not true, as it is often the case for most natural languages, then a syntax tree with more nodes will be less likely to receive a high score because of the multiplication of the large number of rule probabilities. ArchTran overcomes these problems in two ways [Su *et al.*, 1990a].

First, "context" is defined in terms of a set of statistically measurable events called *phrase levels*. A phrase level is a set of symbols which correspond to a *sentential form* in the sentence generation process. In a generalized LR parser, a syntax tree can be described with the sentential forms that is acquired from rightmost derivation. The transition between two successive phrase levels indicates not only which symbol is derived but also under what contextual environment. For example, a generic phrase level $L_i \equiv \{l_A, A, r_A\}$ and its successor $L_{i-1} \equiv \{l_A, X_1^k, r_A\}$ can always be interpreted as a context-sensitive derivation: $l_A\, A\, r_A \stackrel{+}{\Rightarrow} l_A\, X_1^k\, r_A$ where $A \stackrel{+}{\Rightarrow} X_1 \cdots X_k$ is the derivation, and l_A, r_A are the context under which the derivation takes place. Therefore, the basic context-sensitive model for syntactic score is:

$$\begin{aligned} S_{syn} &\equiv P(Syn|Lex, Words) \equiv P(L_1^m|c_1^n, w_1^n) \\ &\equiv \prod_j P(L_j|L_1^{j-1}, c_1^n, w_1^n) \\ &\approx \prod_j P(L_j|L_{j-1}) \\ &\approx \prod_j P(\{l_A, A, r_A\}|\{l_A, X_1^k, r_A\}) \end{aligned} \quad (2)$$

The transition probabilities between phrase levels can be evaluated by considering full context. Hence, arbi-

trary degree of context-sensitivity can be achieved with a context-free grammar with the above formulation. However, for simplicity in evaluation and for the local property of the context, only a finite window size around the reduced (or derived) symbols needs to be considered.

Second, to relieve the normalization problem, the basic context-sensitive model is improved by considering the correlation among phrase levels. In the above model, each term in the syntactic score is evaluated when a **reduce** action is executed. Since the **reduce** actions between successive **shift** actions are highly correlated, we can jointly consider these correlated phrase levels as a single event, and express syntactic score in terms of the phrase levels that correspond to **shift** actions. The syntactic score will then be given as

$$S_{syn} \approx \prod_j P(L'_j|L'_{j-1}) \qquad (2')$$

where $\{L'\}$ is the set of phrase levels whose "prefix" contains the set of symbols in the pushdown stack after an input symbol is shifted to the stack. In other words, they stand for the snap-shots of the parser configuration immediately after each new input symbol is fetched.

The number of such phrase levels, and hence the number of transition probabilities, is always the same for all ambiguous constructions because the number of input tokens for a sentence is fixed. Therefore, the normalization problem can be relieved, and the partial score can be evaluated at each word boundary by monitoring the changes in stack configuration. It provides a way to consider both *intra-level context-sensitivity* and *inter-level correlation* of the underlying context-free grammar.

3.1.3 Semantic Score

Probabilistic semantic models are even harder to construct than lexical and syntactic scores. Traditional approaches to ambiguity resolution in rule-based systems are mainly based on a large number of constraints specified in the form of feature co-occurrence or selectional restriction. The problem with these paradigms lies in several facts:

1. The semantic hierarchy used to specify linguistic rules is large and hard to be handled consistently by human. It is also not clear which of these primitives are discriminant for encoding "semantics".
2. The ambiguity resolution problem is resolved practically on a problem-by-problem basis with different primitives and different mechanisms. There is basically no uniform mechanism for handling general ambiguity resolution problems.

As a result, many rule-based systems adopt "full-blown" semantic analysis for ambiguity resolution. A general belief is that as long as a large amount of fine-grained knowledge is added to the system, the performance of the system will be improved. From our engineering experiences, however, such belief is true only for restricted domain and for the known training text.

Although there are no universally acceptable formalism about semantics, and it is not clear how the large number of semantic primitives interact with each other, we believe that a well formulated feature-based system with discriminant features as the semantic primitives is appropriate for the general problem of ambiguity resolution in the translation task based on our experiences in other disciplines.

Since the main task for semantic analysis is to provide discriminant information for disambiguation, the ArchTran formulation focuses its attention on several issues:

1. Encode the semantic objects of an interpretation with *discriminant* features so that redundant information for disambiguation can be discarded.
2. Encode feature co-occurrence and selectional restriction in probabilistic sense as the linguistics basis for disambiguation.
3. Provide a uniform mechanism for the resolution of general ambiguity problems.

These goals are realized in several ways. First, only *major categories* are used to compose the semantics of each constituent in a sentence. By major categories, we mean verbs, nouns, prepositions, adjectives and adverbs, which carry most of the semantic information of a sentence. Function words, which do not carry much semantic information, are not used in semantic representation.

Second, the semantics is encoded as a form of annotation to the syntax trees; each node is annotated with a feature structure. To simplify the encoding scheme, an ordered N-dimensional feature vector, called semantic N-tuple, is used, instead of the complex feature structure, to characterize the compositional semantics of a constituent. A semantic N-tuple consists of features, in decreasing order of importance, that are used to characterize a constituent. The first component, being the most important one, is called the "head feature" of the semantic N-tuple. A mother node takes its semantic N-tuple from its children by filling the j th component with the "head feature" of its "jth head". By the "jth head", it refers to the child which is ranked at the j th place when considering its contribution to the compositional semantics of its mother node.

Such encoding scheme tactically simulates the feature percolation process of a unification-based mechanism in probabilistic domain, and retains the general characteristics of compositionality of semantics. With such a formulation, the semantic features of a particular interpretation can be characterized by the annotated nodes in the syntax tree. The semantic score can thus be expressed as:

$$\begin{aligned} S_{sem} &\equiv P(Sem|Syn, Lex, Words) \\ &\equiv P(\Gamma_1^m|L_1^m, c_1^n, w_1^n) \\ &\equiv \prod_j P(\Gamma_j|\Gamma_1^{j-1}, L_1^m, c_1^n, w_1^n) \\ &\approx \prod_j P(\Gamma_j|\Gamma_{j-1}) \\ &\approx \prod_j P(\{\overline{l}_A, \overline{A}, \overline{r}_A\}|\{\overline{l}_A, \overline{X}_1^k, \overline{r}_A\}) \end{aligned} \qquad (3)$$

where Γ_j stands for the j th *annotated* phrase level, whose elements, such as $\overline{A}$, are identical with the corresponding phrase level but with semantics annotation.

The formulation above provides a way to deal with the general ambiguity resolution problems with a uniformed formulation [Chang, 1990]. It can also be reduced to

specialized probabilistic semantic models for resolving particular ambiguity problems of interest, such as the prepositional phrase attachment problem [Liu, 1989, Liu *et al.*, 1990].

The score function shown above is currently used to improve preference assignment in scored truncation. The results show that a truncation strategy based on the score function is appropriate for reducing the searching space while retaining good translation quality.

3.2 Adaptive Learning and Robustness Issues

The above models serve well in characterizing most of the important features of languages and most information we may need in resolving the translation problems. If the corpus is large enough, the estimated parameters will show high degree of robustness in dealing with input text outside the training corpus. Unfortunately, there are occasions when large corpora are not readily available in comparison to the complexity of the real-life language problems. In these cases, statistical techniques have to be developed in order to overcome the *sparse data problem*, and ensure the robustness and discrimination power of the probabilistic knowledge bases.

The problems of acquiring more reliable estimation can be explored in several ways. One can use the bootstrap technique [Efron, 1979] to get a better estimation of the probabilities. The smoothing techniques, as in the example of lexical score evaluation, can also be used to assign probabilities to null entries, or to assign different weights to databases of different degree of reliabilities.

In addition to improving the probabilistic databases by improving the methods of estimation, an adaptive learning procedure is required to adjust the estimated parameters, according to the misjudged instances or unreliably recognized instances. The adjustment must satisfy some desirable properties. In particular, the enhencement of *discrimination* and *robustness* of the probabilistic databases is emphasized [Su and Lee, 1991a]. The reason for adjustment is as follows.

In the preceding models, the recognition is achieved indirectly through the maximum likelihood estimation of the model parameters of the training corpora. However, the recognition is actually related to the real ranks, not the estimated values, of the competing analyses. Hence, maximizing likelihood in the training corpora is not equivalent to minimizing the error rate in the training corpora. Therefore, we should try to find a discrimination function, $g(\mathrm{O}', \Lambda')$, which can preserve the real ranks of the competing analyses in terms of the (transformed) observation vectors O' and the (adjusted) parameter set Λ'.

The ArchTran approach to the enhencement of the discrimination power of the statistical databases is therefore directed toward (i) adjusting the estimated parameters, (ii) transforming the observation vectors to a vector space with more discrimination power, and (iii) adopting another measuring function that can be more reliably estimated than probabilistic measure.

Furthermore, the training corpora may have statistical variation with respect to the unrestricted text. Hence, minimizing the error rate in the training corpora does not imply that the recognition rate in the unrestricted text will also be maximized; the parameter set must be robust enough to take care of such statistical variations. This is done in ArchTran by increasing the degree of *separation* (in terms of some distance measure) between the correct analysis and the other competing candidates.

3.3 Probabilistic Transfer Model and Generation Model

Although transfer and generation can be more easily performed once the correct analysis is acquired with the score function, there are still some computational problems with conventional rule-based approaches. Most notably is the problem of identifying the locally transferable units and the canonical sequence of transfer operations. If the probabilistic transfer model can be trained, then the laborious works of finding the mapping can be reduced to a great extent.

The ArchTran approach currently under development is to reduce an annotated syntax tree (AST) to a *normalized* annotated syntax tree (NAST) which consists of locally transferable units. If the reduction process can be done easily, then the transfer score can be divided into individual terms, each of which corresponds to a permutation probability of a locally transferable unit. In other words, we have the following transfer score formulation:

$$\begin{aligned} S_{trf} &\equiv P(T_t|T_s) \approx P(\hat{T}_t|\hat{T}_s) \\ &= \textstyle\prod_i P(p_j([s]_i) \mid [s]_i) \end{aligned} \tag{4}$$

where T_t and T_s are the target and source ASTs, $\hat{T}_t$ and $\hat{T}_s$ are their normalized version, $[s]_i$ is the ith locally transferable unit of the source AST, and $p_j(\cdot)$ is the jth permutation function that transfers $[s]_i$ into its target equivalent ($[t]_i \equiv p_j([s]_i)$).

Given the locally transferred units, we can then find the segments of the generated target text by finding the ones that maximize the following generation score.

$$\begin{aligned} &S_{gen} \approx P(t|T_t) \approx P(t|\hat{T}_t) \\ &= \textstyle\prod_i P(t_i \mid [t]_i) \end{aligned} \tag{5}$$

where t is the target sentence consisting of the segments t_i. In general, any localized transfer units can be used to encode the normalized AST.

4 Flow of Translation and Translation Environment

The flow of translation in ArchTran is divided into seven stages: text preparation, scanner, preprocessor, parser, transfer, synthesizer, and finally the post-editing [Su *et al.*, 1987]. The translation environment of ArchTran is shown in the appendix.

Instead of providing stand-alone machine aided translation systems, our policy is to provide users an environment very much like in-house translation environment. The outcome is a VAN-based translation service center. In the VAN-based services, texts are transmitted from/to the customers, the service center, and the post-editors via a local or public data network. This greatly reduces the overhead of the customers in data

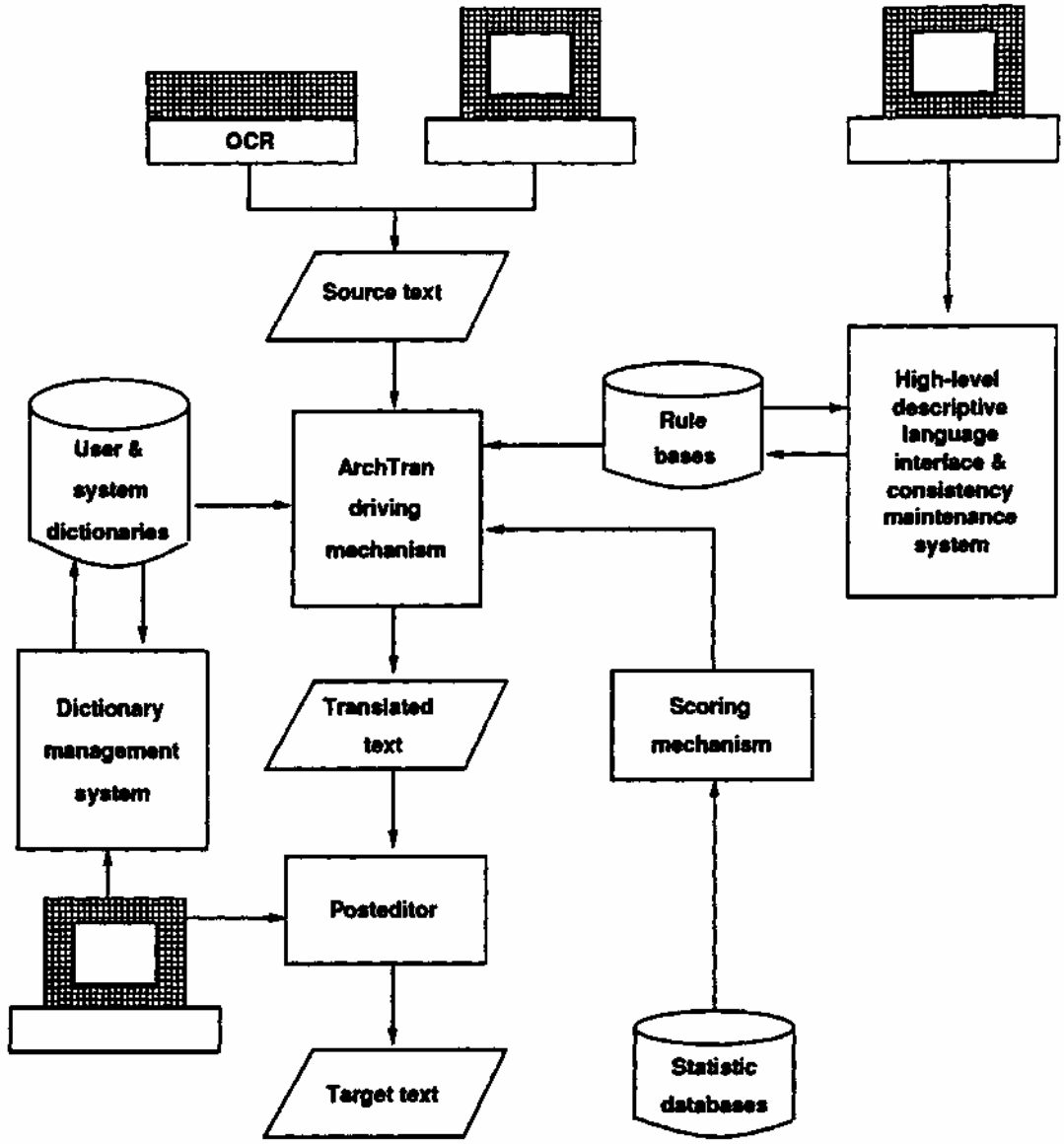

Figure 1: **Translation Environment of Archtran.**

transfer in terms of time, cost, and security assurance. The clients are also relieved of the overhead of maintaining the knowledge bases (e.g. dictionaries) of the system. Sophisticated support tools are also packaged into a translation workstation, integrating aids such as OCR, special-designed text editor, in-house glossaries, bi-texts, DTP, and so on.

5 Future Work

Several techniques are under development to upgrade the current version of ArchTran. These include the research on adaptive learning of language model, error recovery strategies, probabilistic bi-text transfer model, domain identification, domain adaptation, and research on bootstraping and robust techniques for estimating probabilistic model parameters.

References

[Chang, 1990] Chang, J.-S. GPSM: A Generalized Probabilistic Semantic Model for Ambiguity Resolution. Master's thesis, Institute of Computer Science and Information Engineering, National Chiao Tung University, Hsinchu, Taiwan, 1990.

[Chen *et al.*, 1989] Chen, S.-C, M.-H. Wang and K.-Y. Su. A Unification-based Approach to Lexicography for Machine Translation System, In *Proceedings of ROCLING-II*, Nantou, Taiwan; pages 147–161, 1989.

[Church, 1988] Church, K., A Stochastic Parts Program and Noun Phrase Parser for Unrestricted Text, In *ACL Proc. 2nd Conf. on Applied Natural Language Processing*, pages 136–143, Austin, Texas, USA, 9–12 Feb. 1988.

[Church and Hanks, 1989] Church, K. and P. Hanks, Word Association Norms, Mutual Information, and

Lexicography, In *Proc. 27th Annual Meeting of the ACL*, pages 76–83, University of British Columbia, Vancouver, British Columbia, Canada, 26–29 June 1989.

[DeRose and Steven, 1988] DeRose, Steven. J., Grammatical Category Disambiguation by Statistical Optimization, *Computational Linguistics*, 14(1):31–39, 1988.

[Efron, 1979] Efron, B., Bootstrap Methods: Another Look at the Jackknife. *The Annals of Statistics*, 7(1): 1–26. 1979.

[Fu, 1982] Fu, K.-S. *Syntactic Pattern Recognition and Applications*. Prentice-Hall, Englewood Cliffs, NJ, U.S.A., 1982.

[Garside *et al.*, 1987] Garside, Roger, Geoffrey Leech and Geoffrey Sampson (eds.), *The Computational Analysis of English: A Corpus-Based Approach*, Longman Inc., New York, 1987.

[Hsu and Su, 1986] Hsu, H.-H., and K.-Y. Su. A Bottom-Up Parser in The Machine Translation System with the Essence of ATN, In *Proceedings of International Computer Symposium ICS*, Vol. 1 of 3, pages 166–173, Tainan, Taiwan, 1986.

[Liu, 1989] Liu, C.-L. On the Resolution of English PP Attachment Problem with a Probabilistic Semantic Model. Master's thesis, Department of Electrical Engineering, National Tsing Hua University, Hsinchu, Taiwan, 1989.

[Liu *et al.*, 1990] Liu, C.-L, J.-S. Chang and K.-Y. Su. The Semantic Score Approach to the Disambiguation of PP Attachment Problem, In *Proceedings of ROCLING-III*, pages 253–270, Taipei, Taiwan, 1990.

[Su *et al.*, 1987] Su, K.-Y., J.-S. Chang and H.-H. Hsu, A Powerful Language Processing System for English-Chinese Machine Translation, In *Proc. of 1987 Int. Conf. on Chinese and Oriental Language Computing*, pages 260–264, Chicago, IL, U.S.A., 15–17 June 1987.

[Su and Chang, 1988] Su, K.-Y., and J.-S. Chang. Semantic and Syntactic Aspects of Score Function, In *Proceedings of COLING-88*, pages 642–644, Budapest, Hungary, 1988.

[Su *et al.*, 1989] Su, K.-Y, J.-N. Wang, M.-H. Su and J.-S. Chang. A Sequential Truncation Parsing Algorithm Based on the Score Function, In *Proceedings of International Workshop on Parsing Technologies*, pages 95–104, Pittsburgh, U.S.A., 1989.

[Su *et al.*, 1990a] Su, K.-Y., J.-N. Wang, M.-H. Su and J.-S. Chang. GLR Parsing with Scoring. To appear in M. Tomita (ed.), *Generalized LR Parsing*, 1990.

[Su and Chang, 1990b] Su, K.-Y., and J.-S Chang 1990. Some Key Issues in Designing MT Systems, *Machine Translation*, 5(4):265–300, 1990.

[Su and Lee, 1991a] Su, K.-Y., and C.-H. Lee, 1991. Robustness and Discrimination Oriented Speech Recognition Using Weighted HMM and Subspace Projection Approach, In *Proceedings of IEEE ICASSP-91*, Vol. 1, pages 541–544, Toronto, Ontario, Canada. May 14-17, 1991.

[Su *et al.*, 1991b] Su, K.-Y., J.-S. Chang and Y.-C. Lin. 1991. A Unified Approach to Disambiguation Using A Uniform Formulation of Probabilistic Score Functions. In preparation.