Combining Linguistics and Statistics for

High-Quality Limited Domain English-Chinese Machine Translation

By

Yushi Xu

B.Eng in Computer Science (2006)
Shanghai Jiaotong University

Submitted to Department of Electrical Engineering and Computer Science
in Partial Fulfillment
of the Requirements for the Degree of Master of Science

at the

Massachusetts Institute of Technology

June, 2008

Signature of the author ______________________________
Department of Electrical Engineering and Computer Science
June, 2008

Certified by ______________________________
Stephanie Seneff
Principal Research Scientist
Thesis Supervisor

Accepted by ______________________________
Terry P.Orlando
Chair, Department Committee on Graduate Students

Combining Linguistics and Statistics for

High-Quality Limited Domain English-Chinese Machine Translation

By

Yushi Xu

Submitted to Department of Electrical Engineering and Computer Science
On May 9, 2008 in Partial Fulfillment
of the Requirements for the Degree of Master of Science
in Electrical Engineering and Computer Science

ABSTRACT

Second language learning is a compelling activity in today's global markets. This thesis focuses on critical technology necessary to produce a computer spoken translation game for learning Mandarin Chinese in a relatively broad travel domain. Three main aspects are addressed: efficient Chinese parsing, high-quality English-Chinese machine translation, and how these technologies can be integrated into a translation game system.

In the language understanding component, the TINA parser is enhanced with bottom-up and long distance constraint features. The results showed that with these features, the Chinese grammar ran ten times faster and covered 15% more of the test set. In the machine translation component, a combined method of linguistic and statistical system is introduced. The English-Chinese translation is done via an intermediate language "Zhonglish", where the English-Zhonglish translation is accomplished by a parse-and-paraphrase paradigm using hand-coded rules, mainly for structural reconstruction. Zhonglish-Chinese translation is accomplished by a standard phrase-based statistical machine translation system, mostly accomplishing word sense disambiguation and lexicon mapping. We evaluated in an independent test set in IWSLT travel domain spoken language corpus. Substantial improvements were achieved for GIZA alignment crossover: we obtained a 45% decrease in crossovers compared to a traditional phrase-based statistical MT system. Furthermore, the BLEU score improved by 2 points.

Finally, a framework of the translation game system is described, and the feasibility of integrating the components to produce reference translation and to automatically assess student's translation is verified.

Thesis Supervisor: Stephanie Seneff

Title: Principal Research Scientist

Acknowledgments

I would like to thank my supervisor Stephanie Seneff for her patient guidance, wonderful suggestions and generous encouragements in the progress of this thesis. I would also like to express my gratitude to Chao Wang, who had been my co-supervisor before she left MIT. Her kind help enabled me to have a smooth start of the research in this area.

I would like to thank Victor Zue and all the members in Spoken Language Systems Group for creating such a delightful research environment and providing various help at any time. Thanks also go to Peter Sullivan and other students from the UROP program who helped in investigating and building the hand-coded rules in this thesis.

I am thankful to all my friends, both in China and in the US, who helped me get through my first experience of living abroad alone.

Finally, I would like to give my gratitude to my parents. Without their love and support, I would not be able to study abroad in such a great school.

This thesis was supported by MIT and the Delta Electronics Environmental and Educational Foundation.

Table of Contents

Chapter 1. Introduction

In today's global markets, second language learning is becoming a compelling activity to enhance people's skill set. While many more people hope to, or are required to, become proficient in communicating using a second or even a third language, the teaching and learning methods for second language acquisition have changed little over decades. The main sources of learning remain the classroom and textbook, with the teacher standing in the front, explaining some difficult grammar points and assigning written homework. Whether this is a good and "correct" way of teaching a second language is arguable, but it notably gives students little chance to actually speak in the language. The severe shortage of teachers or partners to listen to the students speaking clearly provides an opportunity for fast-developing computer technology to help with this situation. If the computer is capable of interacting with the student, acting as a language teacher, it will be affordable any time and anywhere. It is infinitely patient and it has some complementary features in its tendency not to criticise or be judgmental.

This thesis will focus on critical technology necessary to produce a computer spoken translation game for learning Mandarin Chinese. In the game system usage, the student is given a randomly generated set of English sentences, organized from easy to difficult, and is asked to speak a sentence in Mandarin of equivalent meaning. The system then judges whether the translation is correct or not, well-formed or not, and gives the student feedback according to its judgment. The judgments are made by comparing meaning representations rather than strings. The student can also have access to the reference translation generated automatically by the system. There are two main core components in the system. One is the language understanding component. The system parses the student's translation to see if it's grammatical, and then turns it into a meaning representation to see if it's correct. The other is the machine translation component. The system needs to provide a reference translation for each English sentence.

This system is challenging in two aspects. Firstly, it is about two languages with tremendous differences. The word orders of English and Chinese are substantially different. The most obvious example is wh-questions. In English, the wh-movement is overt, which means that wh-words (what, who, when, etc.) move to the front; whereas in Chinese, they stay in the same positions as their non-wh counterparts. Also, in English, prepositional phrases, adjective clauses, etc. are positioned to the right of the word they modify; in Chinese, they are usually to the left of the word. Furthermore, Chinese has many unique constructions and disjunctive phrases. All these pose difficulty for the traditional phrase-based statistical machine translation approach. The other challenge is that the system is going to work with a relatively open domain -- the travel domain. The corpus we use to build the system is the so-called IWSLT[1] corpus, which includes around 40,000 pairs of parallel sentences. This domain includes a broad range of topics such as flight reservations, shopping, dining, city navigation, tourist information, etc. The sentences are less restricted in terms of structure and vocabulary than those used in previous research conducted in the SLS group [Wang and Seneff, 2007; Chao et al, 2007].

The two challenges result in the requirement of a more powerful language understanding and generation component, both to parse the input Mandarin from the student and to provide a high-quality reference translation for the student. A core contribution of this thesis will be to explore new approaches to deal with Chinese parsing and high-quality translation using a combined method of linguistic and statistical machine translation.

[1] International Workshop of Spoken Language Translation

Chapter 2. Previous Research

This research builds on two Web-based Mandarin learning systems that have been developed previously in our group. One is the translation game in the flight domain [Wang and Seneff, 2007], including flight query and reservation. The system can be accessed by a URL. After login, the web interface shows a set of English sentences for the users to translate into Mandarin Chinese. The sentences are generated by templates reflecting the current difficulty level. The system assumes the user knows very little Chinese. The sentences for the first level are very short and simple. Then gradually, sentences will become longer and the structures will become more complex. The user can speak through a microphone to provide his/her translation, or can click a button to hear a "tutor" provide a correct translation of the sentence if he/she needs help. After the system receives the user's translation by the speech recognizer, it judges the correctness of the translation by the following steps. First, it generates a reference translation by a formal parse-generation method. Then, the reference translation and the user's translation are transformed into meaning representations. Key-value pairs are extracted from the two meaning representations and are compared. If the comparison is successful, the system will congratulate the student and move on to the next sentence. After all the sentences in the current level are translated, the system will judge whether to change the level up or down, or stay at the same level, according to the student's performance.

The other Mandarin Chinese learning system has a similar framework, but its domain involves hobby and activity scheduling [Chao et al, 2007]. The interface is a little different. Instead of prompting the sentences one by one, it displays all the sentences in the current level on the screen at one time. The user is allowed to use any order to translate them. He/she doesn't need to specify the order by clicking the sentence that is going to be translated or in any other way. The system will automatically recognize which sentence the user is translating and give an appropriate response. In this way, the user can get a better feeling that the system is understanding his/her words and is interacting with him/her.

Despite the differences in the domain and web interface of the two Chinese learning systems, the technologies inside are very similar. Both systems adopt a formal parse-generation method, which is supported by two natural language processing systems – TINA [Seneff, 1992] and GENESIS [Baptist and Seneff, 2000]. TINA is a language understanding system designed specifically for spoken language. It utilizes a context-free grammar to define grammar patterns, together with some non-context-free mechanisms to enforce long-distance constraints. These mechanisms include trace, which can restore movements in the surface string, and verb argument constraints, which can select different arguments for different verbs. The main function of TINA is to parse a sentence, produce a parse tree and optionally turn the parse tree into a hierarchical meaning representation. On the other hand, the language generation system GENESIS, together with its preprocessor PLUTO [Cowan, 2004] does the reverse processing. It generates a surface string from a meaning representation. The GENESIS system operates with rule templates. The rules dictate the order in which the constituents are to be generated. The three basic constituents in a semantic frame, namely, clauses, topics (noun phrases), and predicates (broadly including verb phrases, prepositional phrases, and adjective phrases) each have a default generation rule template, which handles the majority of the constituents. In addition, every specific constituent can have its own unique generation rule, or can be a member of a special class which shares a unique generation rule. Both TINA and GENESIS can be adapted to any language as long as the corresponding set of grammar rules are provided.

The two language learning systems described above have several advantages. First of all, they're easy to access. The Internet has become widely available, and a microphone is easily affordable. Also, a Web-based model eliminates annoying installation. Secondly, they provide a way for language learners to speak and communicate in the target language. The chances that the student can practice speaking Chinese by using these two systems are much greater than in a traditional Chinese classroom. And the system's response is highly accurate in terms of both reference translation and automatic assessment of student's performance. User studies of both systems showed that students like the systems and think they're helpful.

However, both systems are developed within a very limited domain. The sentence patterns and vocabulary are very constrained in both the flight domain and the hobby domain. This simplifies the implementation of the system. Manual linguistic rules play a major role in both systems. They provide fast and accurate results in the limited domain. But the limited domain also constrains the helpfulness of the system. To enable the students to learn spoken Mandarin of larger variety, the domain must be less restricted, such as the IWSLT domain. The IWSLT corpus is collected from travelers from different countries. The sentences cover all kinds of situations a traveler may encounter, and a high percentage of the sentences are wh-questions. Compared with the flight domain and the hobby domain, the sentence structures in the IWSLT domain are of much greater variety and the vocabulary is much larger. To deal with such a domain, the parser needs to be more efficient because the grammar will be much more complex. And statistical methods, specifically statistical machine translation can play a larger role in providing the reference translation.

A statistical machine translation system is a system that uses probabilistic models to convert a source language sentence into a target language sentence. The parameters of the models are learned automatically from a bilingual corpus. One typical example is a phrase-based statistical machine translation system MOSES [Koehn et al., 2007]. In the past several decades, statistical machine translation systems have shown their power in learning lexicon mappings and translation models with very little human effort. Statistical machine translation systems outperform linguistic systems in many tasks such as translating news corpora.

But in order to acquire good performance, the statistical systems usually need large amounts of training data, for example one million parallel sentences. And also they're not good at long distance reordering, because they learn from the word sequence only. They don't have the knowledge of the underlying syntax structure of the input sentence. Both defects are fatal here. As a system for spoken language, there doesn't exist as much bilingual data as for written language. The IWSLT corpus we use in this thesis only contains about 40,000 parallel sentences. And, long distance reordering occurs very frequently when translating between two largely different languages, such as Mandarin and English. The large percentage of wh-questions in the IWSLT domain further emphasizes this issue. So basic statistical methods do not suffice. Linguistic knowledge must also be combined into the system. A number of researchers have explored this idea on the news corpus. There is research that uses the parse trees on the source-language. [Yamada and Knight, 2001] did tree-to-string translation to directly transform the source-language parse tree into the target-language string. [Zhang et al., 2007] performed statistical phrase reordering based on the source-language parse tree. [Wang et al., 2007] tried to reorder the tree nodes using linguistic rules. There are also systems that use syntactic knowledge on the target-language side [Marcu et al., 2006]. Research has also been conducted in the IWSLT domain [Shen et al. 2006; Shen et al, 2007; Carpuat and Wu, 2007]. But most of this work is done for Chinese-English translation. Little has been done in the reverse direction.

In the remainder of this thesis, Chapters 3 and 4 will describe two core technologies for the thesis. Chapter 3 will talk about parsing Chinese and Chapter 4 will talk about the combined linguistic-statistical English-Chinese machine translation for general purposes. Chapter 5 will discuss the game configuration that will utilize the new technologies. We conclude with a summary discussion in Chapter 6.

Chapter 3. Parsing Chinese

Being able to parse the Chinese sentences correctly is one important component for a Chinese learning system. When the student inputs his/her Chinese translation, the system needs to understand the sentence and give appropriate feedback. The first step of understanding is parsing. The two existing language learning systems each come with their own domain-specific Chinese grammar which can be loaded by TINA. However, those grammars were developed for very limited domains. The rules and vocabulary did not cover many sentence patterns. There is another Chinese grammar previously developed in our group that is targeted at a broad domain. The initial coverage of this grammar on an IWSLT corpus of 23,768 sentences was around 60%. But when trying to extend the grammar to include more sentence structures, the parsing process became very slow, and the number of sentences which failed due to the consequence that the search space grew too large during parsing increased greatly. The main reason, after looking into the parsing process step by step, is that TINA is designed for English, and Chinese differs from English in a number of ways in terms of the grammar structure. In this chapter, I will talk about features that were introduced into TINA to deal with such different grammar structures. In the last section of the chapter, I will conclude by quantifying the improvements gained with the new features.

3.1 Restructure: A Bottom-Up Strategy on a Top-Down Paradigm

3.1.1 Look Left and Absorb

The core of TINA is a top-down parser. When it sees a word, it tries to predict all the possible phrases that the word together with the current state can lead to. This strategy corresponds to head-initial languages such as English. In English, the most important word of a phrase, or the head, is usually at the beginning of a phrase. For example, in Figure 3-1 phrases a) and b) have the same prepositional phrase "in the room". But when the parser sees "students", it can immediately predict that this is a noun phrase, and when it sees the word "read", it can predict that this is a verb phrase.

However, the situation is different when it comes to Chinese. Although Chinese and English share the same SVO structure, they have distinct word orders for modifiers like prepositional phrase. Phrases c) and d) in Figure 3-1 are the translations for phrases a) and b) respectively. The modifying prepositional phrase now preceeds the noun or the verb. When the parser sees the first word "在", it has to make two predictions: the phrase can either be a noun phrase or a verb phrase. Both choices contain the prepositional prefix. When the second word "房间" comes in, the parser will have to advance both hypothesis. So will it for the third word. The result is that it actually has parsed the exact same prepositional phrase twice.

a) students in the room
b) read in the room
c) 在 房间 里 的 学生
at room in DE student
d) 在 房间 里 读书
at room in read

Figure 3-1Different Positions of PP in English and Chinese

The problem may not be noticeable in the above example, but when the common prefix is long and complicated, it will consume large space and time. And this is frequently the case in Chinese. Consider the sentences in Figure 3-2. Sentences a) and b) show how an adverbial clause can be formed in Chinese by adding a conjunction at the end of the clause. Sentences c) and d) show how a subject clause can be formed. In both examples, the clauses are in the exact form of a complete sentence. Figure 3-3 shows a set of possible grammar rules that can describe this kind of structure without introducing left-recursion. But it is obvious that the parser needs to redundantly parse the NP VP construct three times because it doesn't know whether it is a complete sentence, an adverbial clause or a subject clause until it reaches the word after the VP. What makes the situation worse is that these rules are at the very top level. This means that, for every input sentence, the parser will do redundant work at least three times. Together with all the other rules, the result is that the parser spends lots of time and space on redundant parsing.

a) 他 会 开车
he can drive
He can drive.

b) 他 会 开车 的话 我 就 去
he can drive if I will go
I will go if he can drive.

c) 在 道路 的 右边 行驶
at road NOM right drive
Drive on the right of the road

d) 在 道路 的 右边 行驶 很 不同
at road NOM right drive very different
Driving on the right of the road is very different.

Figure 3-2 More examples of Chinese bottom-up characteristics

Sentence -> AdvClause NP VP
Sentence -> SubjClause VP
Sentence -> NP VP
AdvClause -> NP VP Conj
SubjClause -> NP VP

Figure 3-3 A possible set of grammar rules that can generate sentences in Figure 3-2

There are several ways to help the parser correct this bad situation. One way is to use an algorithm similar to depth-first search. The parser only chooses one hypothesis to advance and stores the intermediate parsing solution. When the first hypothesis finishes, the parser backtracks to an earlier state and tries a second hypothesis. In the second hypothesis, it can use the intermediate solution, say word six to word nine forming a noun phrase, instead of parsing again.

But this approach has several defects. First of all, it requires additional space of size mn^2, in which n is the length of the sentence and m is the number of nonterminal symbols in the grammar. For a generic grammar, the size of m is usually larger than n, so the total space needed is of $O(n^3)$. This is not desired if the system needs to be run on devices with small memory. Secondly, the backtracking algorithm is not natural as humans rarely do backtracking when understanding a sentence. Instead, a native Chinese speaker will process Sentence b) in Figure 3-2 like this: when he sees the word sequence "he can drive", the brain parses it as a complete sentence. Then the word "if" comes in. The marker in mind saying "complete sentence" is changed into "adverbial clause". In other words, it is a bottom-up parsing paradigm. But this does not necessarily require that we change the basic parsing algorithm to be bottom-up, which would require extensive modification to the code. There are still plenty of grammar rules in Chinese that suit the top-down mechanism. We need a method to selectively add the bottom-up ability only when it is needed to the current existing top-down parser.

The solution that comes up is to delay the decision of the phrases that share the same prefix. For example, the parser parses the prepositional phrase as a sibling of the noun/verb phrase and later on decide that the PP should be a child of the noun/verb phrase because it appears to the left of the noun/verb phrase. This can be accomplished through restructuring after the parsing is done: the indicator node looks left and absorbs its left sibling[2]. Figure 3-4 shows how the idea of absorption works on Sentence b) in Figure 3-2.

[2] We originally named this mechanism "dummy trace" because it can be viewed as a special case of the existing trace mechanism in TINA.But later, it became part of the restructuring feature, together with the look-up-and-rename mechanism described in the next sub-section.

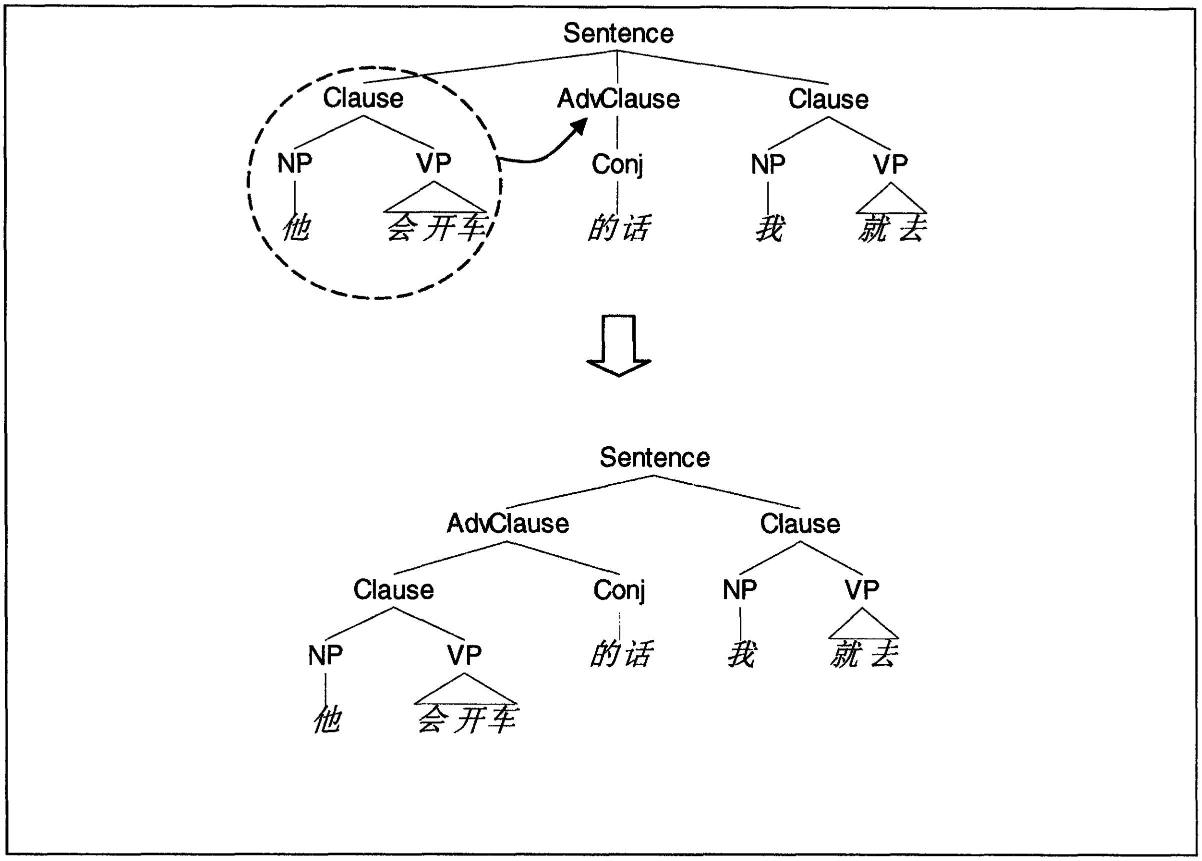


Figure 3-4 How look-left-and-absorb restructuring works on Sentence b) in Figure 3-2

With the look-left-and-absorb restructuring, the grammar needs modification to parse the sentence into a "flat" initial tree. The modification is simply to take out the common prefix of the grammar rules that will cause redundant parsing, and lift the common part to the upper level. The mechanism also needs an additional set of rules to specify the legal actions of absorption. Such a rule contains the indicators, which are the tree nodes whose left sibling should really be considered as their child, and a list of tree nodes that the indicators can absorb. The rule optionally contains the ability to rename the absorbed node. This is useful when dealing with sentences like sentence d) in Figure 3-2, since the grammar doesn't allow an empty grammar rule.

Figure 3-5 exemplifies how the grammar in Figure 3-3 looks after modification. In the right column are the corresponding restructure rules, in which the arrow symbol "->" stands for a rename action. The modified grammar and the restructure rules will eventually produce the exact same tree as the grammar in Figure 3-3, except that it creates an extra top layer for sentences with a subject-clause. But the parsing will be much more efficient this time. When a starting NP comes in, there is only one possible prediction, i.e., it's an NP under Clause and the Clause is under Sentence. The hypothesis will split after the first NP VP sequence is parsed as a Clause, rather than split at the very beginning. After the initial tree is produced, the parser will look for those nodes that should be absorbed by its right sibling according to the restructure rules. The absorption procedure is recursive, so an indicator can absorb a next left sibling after absorbing one, if there are multiple absorbates to its left.

Sentence -> Clause	INDICATOR: AdvClause
Sentence -> Clause AdvClause Clause	ABSORB: Clause
Sentence -> Clause Reclause	
Clause -> NP VP	INDICATOR: Reclause
AdvClause -> Conj	ABSORB: Clause->SubjClause
Reclause -> VP	

Figure 3-5 Modified grammar with restructure support.

3.1.2 Look Up and Rename

The look-left-and-absorb restructure tackled the redundant parsing problem for a language with bottom-up characteristics. But the problem has not been solved completely. Consider the following example in Figure 3-6. This is an example of a Chinese noun clause. By noun clause, we mean a noun phrase which has a clause modifier, and the role of the noun in the clause is the object. We can see that the difference between a normal statement and the noun phrase is only the nominal auxiliary DE before the object. The sentence a) and the phrase b) share exactly the same prefix, including a subject, a temporal, an auxiliary verb and a verb.

a) 我 明天 要 买 书
I tomorrow will buy book
I will buy books tomorrow.
b) 我 明天 要 买 的 书
I tomorrow will buy NOM book
the book I will buy tomorrow

Figure 3-6 Noun clauses in Chinese

To avoid parsing the long prefix twice, we should adopt a strategy of look-left-and-absorb. The grammar rules don't split until after the verb. Then, if there is a nominal auxiliary following the verb, we enter the rule for noun phrase, otherwise we enter the normal verb phrase. However, there are two problems with this approach. The first problem is that we have to make the subject and the object be siblings in order to absorb the subject under noun phrase. This is not the conventional way to parse a sentence. Usually the object is under the verb phrase, and the verb phrase is the sibling of the subject. Another problem is that the restructure is done after parsing, so, during the process of parsing, the verb and the object will not have the same parent. This will block the parser from checking verb argument constraints. The verb argument constraints are an important component to ensure a correct parse. They specify what kinds of arguments a certain verb can take, for example direct object, indirect object, clausal object, etc. The check is only effective for the siblings of the verb. So, if the verb and object are separated into two levels, the constraints have no effect.

So we think again how mentally a Chinese noun clause is parsed in the mind. A Chinese speaker usually starts from expecting a sentence. So the words "I", "tomorrow", "will" and "buy" are comprehended as they are forming an ordinary sentence. But as soon as the nominal auxiliary comes into sight, he/she realizes that his/her expectation is wrong, and revises the mental model that he/she is accepting a noun phrase instead of a sentence from the beginning. To translate this idea into our parsing framework, it is actually to another type of restructure: a special node causes its ancestor node to rename. We call this restructure "look up and rename". Figure 3-7 illustrates using the look-up-and-rename restructure to deal with the noun clauses.

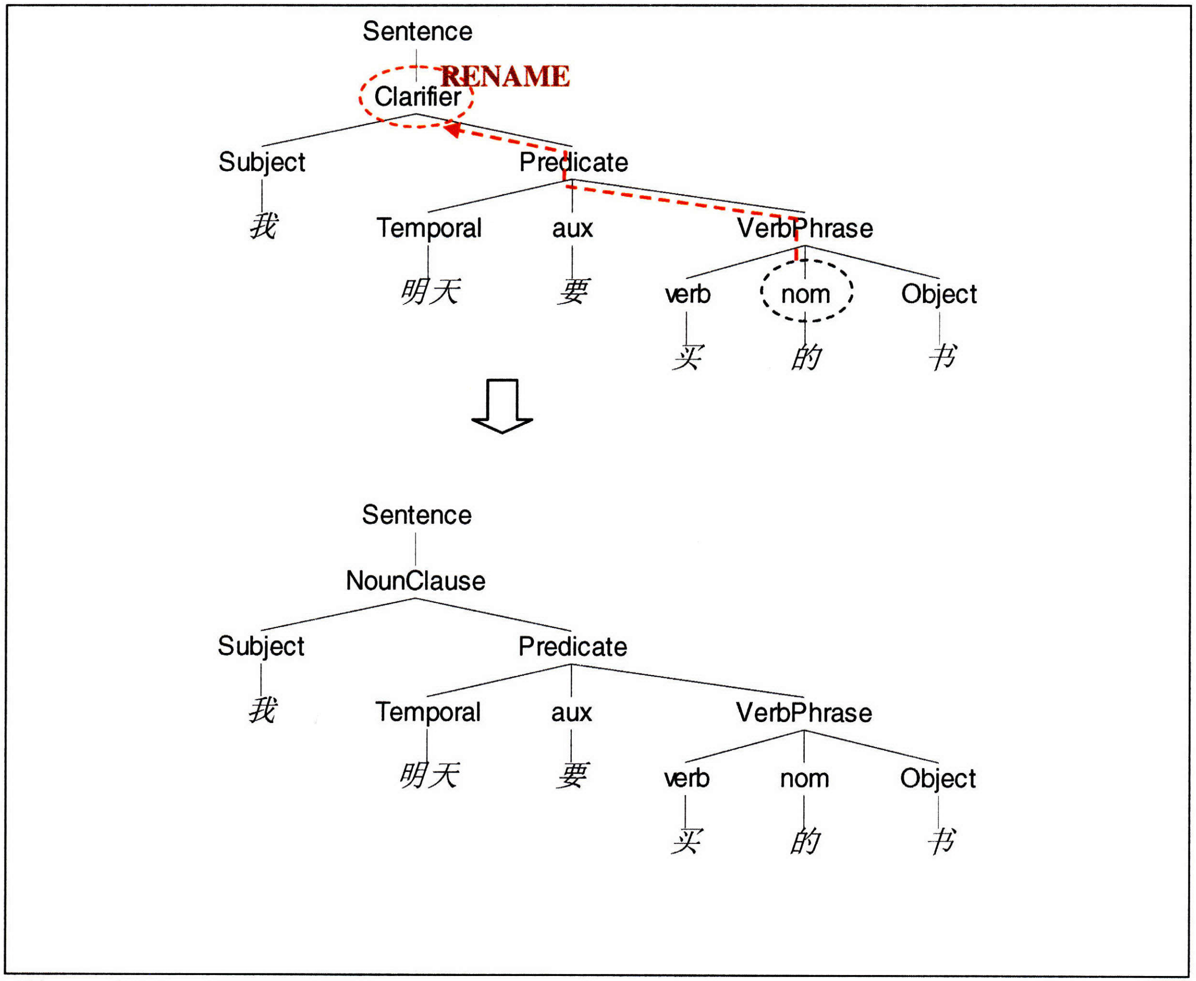


Figure 3-7 Look-up-and-rename restructure for noun clause

With the look-up-and-rename restructure, there is no need for a separate grammar rule for the noun clauses. They can share the same rule with the verb phrase, which has an optional nominal auxiliary between the verb and the object. Then, later on when the parse tree is created, the nominal auxiliary triggers the restructure, and renames the specified ancestor into noun clause. Figure 3-8 shows a demonstration of the grammar rules together with the restructure rules. The blocker in the restructure rules constrains the indicator from renaming ancestors that are higher than the blocker, so that each nominal auxiliary can only rename the closest clarifier in the example.

Sentence -> Clarifier	INDICATOR: nom
Clarifier -> Subject Predicate	RENAME_ANCESTOR: Clarifier
Predicate -> [Temporal] [aux] VerbPhrase	BLOCK_BY: NounClause
VerbPhrase -> verb [nom] Object	

Figure 3-8 Simplified grammar and restructure rules to deal with noun clause

3.1.3 Implementation

The look-left-and-absorb and look-up-and-rename restructure are implemented together as a final step for regular parsing. The restructure rules are written in a separate file as another new specification to the TINA system. The code iterates over all the tree nodes to find the indicators specified in the rule file from left to right. For the look-left-and-absorb, it iterates from top to bottom, and for the look-up-and-rename, it iterates from bottom to top. The two types of restructuring are performed in turn until there is nothing being changed.

3.2 "Parasites and Enemies": Dealing with Reduplicated Words and Long-Distance Constraints

3.2.1 Terminal Parasites

Another distinct feature in Chinese is that there are a lot of reduplicated words. Many nouns, adjectives, adverbs and verbs can be expressed in a reduplicated way. Figure 3-9 gives some examples.

a) Reduplicated noun:
他 *天天* 上班 。
He goes to work everyday.

b) Reduplicated adjective:
房间 里 *漆黑漆黑* 的 。
It's pitch-dark in the room.

c) Reduplicated verb:
让 我 *看一 看*。
Let me have a look

Figure 3-9 Examples of reduplicated words in Chinese

Reduplicating nouns, adjectives and adverbs have certain constraints. Only a few nouns, adjectives and adverbs can reduplicate. They don't really cause problems, and indeed, there is little to do with them since the tokenizer usually tokenizes the reduplicated word as a single word. But reduplicated verbs are different. Most single-character verbs can have the following three reduplicated forms, and most multi-character verbs can at least have the first form.

- V V

  e.g. 学 学 learn

  休息 休息 have a rest

- V yi V

  e.g. 学 一 学 learn

- V not V

  e.g. 学 不 学 learn? (Asking whether the subject learns, or wants to learn something)

  学 没 学 learn? (Asking whether the listener has learnt something in the past)

  休息 不 休息 rest? (Asking whether the subject rests)

Because of the large number of verbs, we don't want our lexicon to include all of the reduplicated forms. Doing so is bad for generalization too. But a context-free grammar cannot deal with this problem. The CFG will over-generate a lot of invalid phrases, for example "V1 yi V2". We have to use a non-context-free feature to check the two occurrences of the verb.

We introduce a feature called "parasite" into TINA. The parasite node has a special suffix "<parasite>" to indicate that it is parasitic on some other node. Figure 3-10 shows a short grammar with a parasite that can generate the "V yi V" form. The nonterminal *Verb* under *ShortTimePhrase* has a parasite tag <parasite:VerbPhrase>, which means the terminal word under it must match the terminal word of the node *Verb* somewhere to its left and under node VerbPhrase. We call the node *Verb* the host node, and the *VerbPhrase* the environment of the parasite.

```
VerbPhrase -> Verb ShortTimePhrase
ShortTimePhrase -> [yi] Verb<parasite:VerbPhrase>
```

Figure 3-10 A grammar with a parasite

The Chinese grammar utilizes the parasite feature mainly in two places: the short time-phrase, i.e. the first and the second reduplicated form listed above, and the question verb, i.e. the third reduplicated form. In fact, there are more situations where reduplication can occur, and we will expand the rule set to include those in the future.

3.2.2 Node-Parasites and Node-Enemies

The last subsection discussed the reduplicated forms in Chinese. To think of them in another view, they are actually a problem of disjunctive equality constraint that happens on the terminal words. Besides the reduplicated words, Chinese is also known for many other disjunctive structures and long-distance constraints. Figure 3-11 gives an example with the versatile auxiliary DE. DE is most commonly used as a nominal auxiliary, but it can be used as a tone auxiliary to emphasize some part of the sentence. In the two examples, the existence of DE emphasizes the location of an action in the past. Although the DE in sentence a) can also be interpreted as a nominal auxiliary, in which case the sentence means "the meal I ate at school", this interpretation is not as strong as the other one if these words are the only words in the sentence. However, without the locative prepositional phrase, the interpretation as a noun clause becomes stronger. So we can conclude that DE as an emphasizer depends on some other syntax nodes. This is a similar problem as the reduplicated forms, but only that this time instead of the parasite and the host nodes having the same terminal words, the parasite node is happy enough as long as the host node appears in the environment. To distinguish this new parasite style from the above one, we call it node-parasite.

a) 我在学校 吃 的 饭
I at school eat DE meal
It is at school that I ate my meal.

b) 你 在 哪 吃 的 饭 ?
you at where eat DE meal ?
Where did you eat your meal?

Figure 3-11 Some disjunctive structures and long-distance constraints in Chinese

The function of node-parasite, if expressed in a straightforward way, is that "I cannot live without some certain nodes". So immediately, we think of the opposite situation: "I cannot live together with some certain nodes". And it turns out that this is also a popular constraint in Chinese. Let us take a look at Sentence b) in Figure 3-11. Compared with Sentence a), the main difference is that the location school is substituted by the wh-word "where". But this substitution eliminates the alternative noun clause interpretation. The DE in Sentence b) can only be interpreted as an emphasizer. To use the wording at the beginning of this paragraph, the nominal auxiliary DE cannot live together with the word "where". This feature is named as node-enemy.

Because a node can have several hosts and/or several enemies, we cannot specify the constraints directly in the grammar as we did for terminal parasites. Instead, the node in focus, the environment node and the list of hosts/enemies are specified in a separate file, where other non-context-free constraints are specified, as shown in Figure 3-12. Both the node-parasite and node-enemy add more power to the parser. Because the constraints are cross-level, it also make it easier for the grammar writer. We can care less about some special words in the high level. General categories can be used in high levels to incarnate the overall syntax structure. Words of special cases can be subcategories which can be selected by node-parasites or killed by node-enemies.

```
.*node-parasites*
Emphasize_de   VerbPhrase      locative temporal
...

.*node-enemies*
Nominal_de     VerbPhrase      where when how
...
```

Figure 3-12 Syntax of specification for node-parasite and node-enemy

3.2.3 Implementation

The parasite and enemy features are implemented in the constraint checking part of TINA. The checking is performed during the regular parsing process, along with other non-context-free constraints. The algorithm is very straightforward. We iterate over all the nodes under the environment node. If a parasite is found, we succeed once we have found a qualified host node. If it is checking for an enemy, the current parsing theory will be killed upon identification of an enemy.

3.3 Quantifying Improvements

Evaluating a grammar is a difficult task. It is hard to judge whether a sentence is parsed correctly by some automatic means. Even human experts may disagree on the judgments. So instead, we use two simple measurements to evaluate the Chinese grammar together with the new features of TINA. One is parse coverage. Since our test data are all good Chinese sentences, the chance that a good sentence is parsed but parsed into a weird tree is small. So the parse coverage can roughly reflect how good the grammar is. Another measurement is parsing time. The time consumed is influenced by both the efficiency of the parser and the organization of the grammar.

We conducted the test using the IWSLT 2005 corpus. All the sentences are broken down into single sentences which contain at most one period, question mark or exclamation mark. This results in a set of 23768 sentences of average length 7.5 words per sentence. We parse the sentences using two versions of TINA. The old TINA is the one in which the restructure and parasite features were originally implemented. Recently, the TINA parsing code has been carefully rewritten to improve speed and efficiency, with the goal of running the system on a mobile device. I will refer to this re-implemented version as "TinyTINA". TinyTINA produces the same result as TINA in principle, but runs faster. On both TINA systems, we tested three grammars: the initial grammar which we start from, the extended grammar in which we devoted much effort to increase the parse coverage, and finally the extended shallow grammar that utilizes restructure and parasites. Table 3-1 shows the results in detail.

	Parse Coverage	Time Consumed by TINA	Time Consumed by TinyTINA	# Times that the stack limit is reached before any parse is found
Initial Grammar	63.7%	2357s	472s	894
Extended Grammar	66.1%	5716s	569s	1167
Extended Shallow Grammar	81.2%	475s	76s	133

Table 3-1 Comparison of parse coverage and time consumed on 23,768 sentences.

From the table it is clear that it costs amazingly less time. Compared to the extended grammar, the extended shallow grammar is 12 times faster for the TINA system. Even with TinyTINA which optimizes computation and memory management, the extended shallow grammar is over 7 times faster. And it covers 15% more sentences.

It should be made clear that the extended shallow grammar is not equivalent to the extended grammar. The extended shallow grammar allows more grammatical patterns. But the extended grammar has faced the dilemma that allowing more patterns will fail many sentences that used to be parsable, because the complex rules exceed the parser's stack memory, and this also slows down the parsing notably. It has become almost impossible to make further improvements. But for the extended shallow grammar, we believe there is still lots of opportunity for improvement.

The improvement in efficiency mainly attributes to the restructure mechanism. Though the algorithm has to go through every tree node, in practice the number of nodes is only about two to three times the sentence length. And the number of restructure indicators is only one eighth of the total nonterminals. The time spent on the restructure is much less than what it requires to redundantly parse the phrases.

Chapter 4. English-Chinese Machine Translation

The second piece needed for the translation game system is automatic English-Chinese translation. Whenever an English sentence is prompted for the student to translate, the system needs to prepare a reference Chinese translation. In the previous two language learning systems, a formal parse-generation approach is used to translate an English sentence into Chinese. Manually written linguistic rules play a major role in the parse-generation paradigm. When the domain becomes more unlimited, the size and complexity of the rules will grow. It is difficult to provide high quality with only manually written rules. A method combining linguistic and statistical machine translation would be more favorable. In this chapter, the translation task will be treated as a general English-Chinese problem. Then in the next chapter, I will discuss how to use this idea to generate high-quality translation specifically for our language learning application.

As mentioned above, the challenge for English-Chinese statistical machine translation lies in the large difference between the two languages. Figure 4-1 shows an example. Not having large enough training data for the domain we are interested in is another problem. On the other hand, for the linguistic method, it is daunting to write all the detailed rules to do tedious things like word sense disambiguation. So, combining the two will let us have the merits of both, and possibly overcome the defects of both.

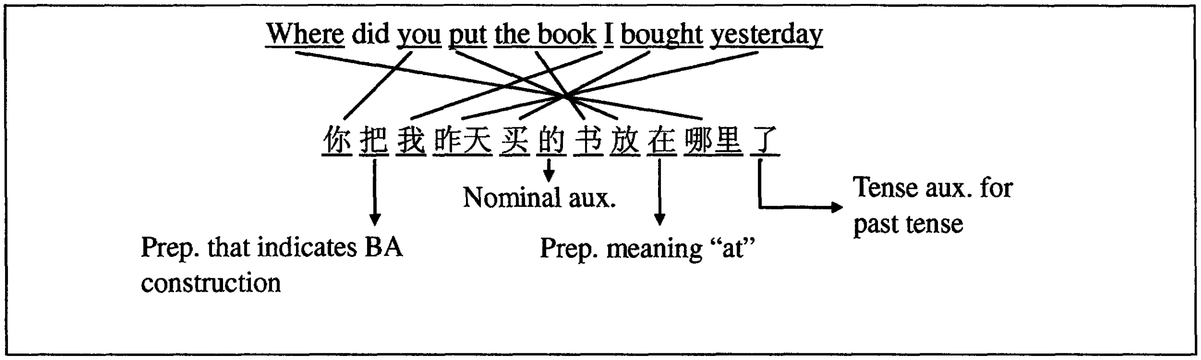

Where did you put the book I bought yesterday
你把我昨天买的书放在哪里了
Nominal aux.
Tense aux. for past tense
Prep. that indicates BA construction
Prep. meaning "at"

Figure 4-1 An example of differences between English and Chinese

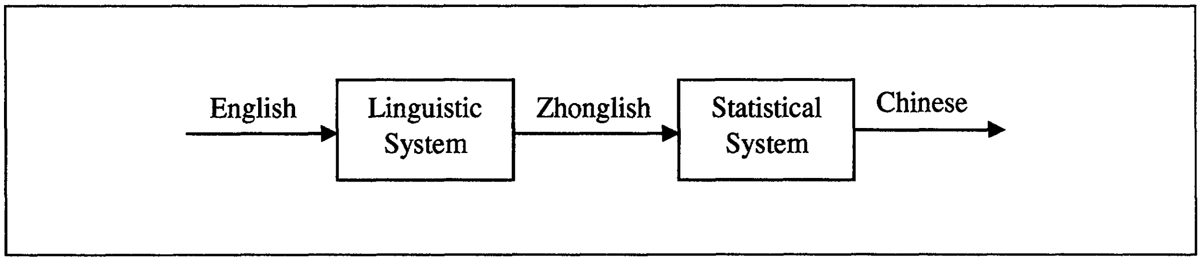


Figure 4-2 High-Level Framework of the Combined Method for English-Chinese Machine Translation

Figure 4-2 shows the high-level framework of how to combine the two methods. The basic idea is to construct a sentence in which the word order is close to Chinese before feeding it into a traditional statistical machine translation system. The constructed sentence maintains most of the words in English, but may have some function words which are unique to Chinese. We call this language "Zhonglish". The English-Chinese translation is done in two steps. First, we use the linguistic system to translate English into Zhonglish. Then, the statistical system further translates Zhonglish into Chinese. In the first step, most of the work is to rewrite the English sentence according to Chinese grammar rules as much as feasible. Little will be concerned about lexicon mapping. Because the rewrite is done by linguistic rules, it is easy to control and re-organize the sentence by constituents. Then, in the second step, a statistical system will translate the Zhonglish into final Chinese. Since Zhonglish and Chinese are very similar in terms of structure, they can potentially be aligned better during training. The statistical system needs to take care of fewer global reorderings, which it is usually poor at, but it will produce a good lexicon, which would take a lot of human effort.

Since the statistical machine translation system we use is a standard phrase-based system, I am not going to describe how it works in detail. In the following sections, I will only talk about how to generate the Zhonglish sentence, various problems we met with during the experiment, and the solutions to those problems.

4.1 Translating English to Zhonglish

The translation from English to Zhonglish is done in the formal parse-generation paradigm, very similar to how the previous two language learning systems generate reference Chinese translations. Two components are involved in this step: the language understanding component, which is TINA specifically, and the language generation component GENESIS. Although both TINA and GENESIS contain a statistical capability to a certain extent, they are classified as linguistic systems on the whole.

The formal parse-generation paradigm works as follows: starting from an English input sentence, TINA parses it using a set of English grammar rules and converts the parse tree into a hierarchical meaning representation. Then GENESIS takes the meaning representation as input, and produces a surface string according to template rules and vocabulary entries. The hierarchical meaning representation, which is called a "semantic frame", serves as an interlingua between the input and output language. It throws away temporal order of the input sentence and only keeps the hierarchical syntax and semantic information. Figure 4-3 shows the simplified semantic frame for the sentence "Where did you put the book I bought yesterday".

```
{c wh_question
  :auxil "xdo"
  :topic {q pronoun
         :name "you"
         :number "pl" }
  :mode "past"
  :pred {p put
        :topic {q object
               :clause_object {c noun_clause
                                   :topic {q pronoun
                                          :name "i"
                                          :number "first" }
                                   :pred {p buy
                                          :topic {q trace_object
                                                 :quantifier "def"
                                                 :noun "book" }
                                          :mode "past"
                                          :pred {p temporal
                                                :topic {q rel_date
                                                       :name "yesterday" }}}}}
        :pred {p trace_pred
               :trace "where" } } }
```

Figure 4-3 Simplified Semantic frame for "Where did you put the book I bought yesterday"

In a semantic frame, the braces represent constituents. There are three type of constituents. Clause, denoted by “c”, refers to statement, verifier, wh-question, noun clause, adjective clause, etc. Topic, denoted by “q” represents the noun phrases. Predicate, denoted by “p” is a broad concept which includes verb phrase, prepositional phrase, adjective phrase, etc. Within each constituent, one or more keys, which start with a colon, supply various kinds of information. The value of a key can be a string, an integer, or another constituent. It is very easy to understand the meaning carried by a semantic frame. For example, the semantic frame in Figure 4-3 represents a wh-question. The subject is “you”, the main verb is “put” and the tense is past. Note the position of “where” and “book” in the semantic frame. Although in the surface string, “where” is separated from the verb phrase by the auxiliary and the subject because of wh-movement, we are able to restore it to its deep syntax structure position, i.e., within the verb phrase. This is powered by the trace mechanism of TINA. The same thing happens to the word “book”. It is restored to the object position of the verb "buy". This is really one of the reasons that the semantic frame works well as an interlingua. When the movement of the two languages differs a lot, for example English and Chinese, the deep syntax structure will make the translation a lot easier, as I will show soon in the next paragraphs.

After we obtain the semantic frame, now it is GENESIS' turn to generate the Zhonglish string. As introduced in Chapter 2, GENESIS works from rule templates. The rules follow GENESIS' syntax, which allows conditional and unconditional branching. Figure 4-4 gives a set of very simple rule templates that can generate English sentences. The clause template generates :topic, which is the subject, and :pred, which is the verb in order. The topic template first generates the quantifier, then the core noun of itself denoted by $core. Lastly, it tells GENESIS to branch into another template called do_prep_phrase to generate the prepositional phrases. Similarly, the predicate template first generates its core verb or preposition, then :topic which should be the object, and finally all the prepositional phrases within the constituents.

```
Clause_template        :topic :pred
Topic_template         :quantifier $core >do_prep_phrase
Predicate_template     $core :topic >do_prep_phrase
```

Figure 4-4Simplified rule templates to generate English

We can see that the syntax is straightforward. If we are going to generate into another language, it can be done by manipulating these rules. And fortunately, our group has already developed a set of high-quality rule templates for English generation. Based on that, here's what we need to do to change it into the Zhonglish generation rules.

Step 1, Undo Overt Movement. Chinese usually retains the wh-phrases in their deep structure positions and we already get this from TINA's trace mechanism. So it is relatively easy to just disable all the sophisticated generation rules that pull the wh-phrases to the front of the sentence. All types of sentences are generated in the style of statements in English.

Step 2, Omit Inflection and Do-Support. Chinese is a morphology-less language. The verbs and nouns don't have morphological changes. Although there are indicators for tense in Chinese, it is appropriate to omit all the inflections of verbs and nouns. Doing so will make the successive statistical system suffer less from sparse data problem. Also, the auxiliary "do" is deleted since there is no such corresponding word in Chinese.

Step 3, Modify the Order of the Constituents and Add Unique Chinese Constructions. As mentioned multiple times above, Chinese differs from English in many places, and this is where most of the work in this part comes from. The generation rules are modified so that the output sentence looks like Chinese. We didn't expect it to follow the Chinese grammar perfectly, but we tried to do so much as possible. We paid special attention to those structures that affect the word order greatly, and inserted some Chinese function words that don't exist in English, like the nominal "DE", questional "MA" and so on. Following is a list of major things we focused on.

Prepositional phrase. Most of the prepositional phrases precede nouns and verbs, but there are some exceptions. For example, prepositional phrases headed by "to" under a verb phrase are usually post-posed. And when the prepositional phrase is an argument rather than an adjunct of a verb, it is post-posed. For example, "stand under the tree" is not expressed as "under the tree stand" in Chinese. We distinguish these cases carefully.

Adjective clause and noun clause. In English, the head noun of both of these two clause types is at the front. In Chinese, the head noun is always at the end, with a nominal auxiliary DE between the modifier and the head noun.

Adverbial clause. Most of the adverbial clauses such as if, while, before, after, etc., precede the main clause. Other adverbial clauses are quite complex. The position of an "until" clause depends on the verb in the main clause, and "as soon as" is translated into the Chinese "YI...JIU..." structure.

There-be structure. There-be structure is a special structure in English which reverses the order of the subject and the verb. A sentence of there-be structure typically looks like "there be subject location/temporal". The Chinese sentence with equivalent meaning is in a special order too, but it puts the location/temporal at the beginning. The Chinese looks like "location/temporal have subject".

We also constructed some unique Chinese constructions. The most typical one is the BA construction. With the BA construction, an SVO sentence is written in the order of S BA O V. BA construction most frequently comes with verbs that have something to do with moving an object, or changing the status of an object, for example "put", "move", "write down". We use this as a clue and make a couple of tests to decide whether to construct the BA construction or not. More will be discussed about the BA construction in the next section.

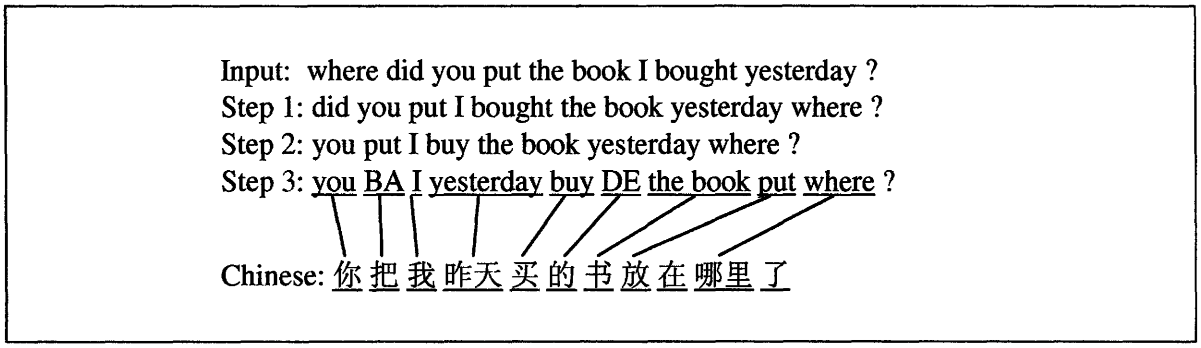
Table 4-1 shows some examples of Zhonglish in comparison with English.

	English	Zhonglish
Prepositional Phrase	A man in the room Work in the room	A in the room DE man In the room work
Adjective Clause	The man reading a book	The read a book DE man
Noun Clause	The book I read	I read DE book
Adverbial Clause	Come in when I call you I will tell her as soon as she comes back	When I call you come in She YI come back I JIU will tell her

There-be structure	There is a book on the table	On the table HAVE a book
Verifier	Can you do it ?	You can do it MA ?
BA construction	Put it on the table	BA it put on the table

Table 4-1 Major changes moving from English to Zhonglish

Through the three steps, an English input will be translated into a Zhonglish output. Figure 4-5 illustrates how the three steps can translate a rather complicated sentence as in Figure 4-1 into fairly good Zhonglish. There are only two function words missing if compared with actual Chinese. Of course, translation sometimes is not as simple as direct word-to-word conversion. In such cases, as long as the Zhonglish is consistent, we can count on the statistical system to learn how to translate them.



Input: where did you put the book I bought yesterday ?
Step 1: did you put I bought the book yesterday where ?
Step 2: you put I buy the book yesterday where ?
Step 3: you BA I yesterday buy DE the book put where ?
Chinese: 你 把 我 昨天 买 的 书 放 在 哪里 了

Figure 4-5 Results from each of the three steps for the sentence "Where did you put the book I bought yesterday"

4.2 Dealing with Random Constructions

In the previous section, I mentioned that we created the BA construction in Zhonglish. The BA construction is considered as an important structure because it reverses the order of the verb and the object, which is not in the range of local reordering. However, as we were trying to determine when to use the BA construction, we realized that we are not able to predict it one hundred percent in correspondence with the reference Chinese. The difficulty is that oftentimes, the BA construction is optional. An example is shown in Figure 4-6. Either translation is good and their meanings don't have any difference. We call this kind of construction "random" because their appearance is not predictable by grammar. In this situation, the two choices are equally good in terms of translation, but in the training process, we really want the one that corresponds to how the parallel Chinese sentence is expressed. We convert English into Zhonglish so that, in the ideal case, the words in Zhonglish align perfectly with the words in Chinese. This will relieve the burden of the statistical system, and potentially generate a better lexicon, phrase table and translation model. So, in order to align well, the choice of using the random constructions or not should be based on the evidence available from the parallel sentence.

Input:	Please open your mouth.
Translation (a):	请 张开 你 的 嘴
	Please open your mouth
Translation (b):	请 把 你 的 嘴 张开
	Please BA your mouth open

Figure 4-6 An example of an optional BA construction

We solved this problem from two aspects. First, we observe the sentences carefully and conclude by designing two sets of tests. One is the conditions when the construction is obligatory. The other set of tests allows for more situation, in which the construction may be generated. Then, we take advantage of the existence of the parallel corpus. When TINA generates the semantic frame for each input English sentence, it also examines the parallel Chinese sentence and sets a flag in the frame to indicate whether the parallel sentence contains the construction or not. The examination is not done by parsing the parallel Chinese sentence, but simply by searching for the keyword of the construction. Take the example of the BA construction: TINA only looks for the character "BA" in the Chinese sentence. If it finds the character, it will set a :has_ba key in the semantic frame. When GENESIS is trying to decide whether to generate the construction, it first checks if the obligatory condition is satisfied. If it is satisfied, then the construction is not “random”, so it will always be generated. Otherwise, if it satisfies the “maybe” condition, the decision is made by looking for the special flag that TINA set in the frame. In this way, we can generate Zhonglish sentences in the same manner as the reference Chinese translations most of the time. The method is fast, because we only do an extra string search. And it is of high accuracy, because the tests will rule out most of the false flags.

In the test process, when there is no parallel sentence to use as reference, the construction is generated in the obligatory condition. For the maybe condition, two candidate Zhonglish sentences will be generated. One has the construction, and the other one doesn’t. Then the language model is used to pick the one with higher score.

4.3 Pre-Translating Phrases

Statistical machine translation systems are powerful. They can learn how to translate phrases and words completely automatically. However, when they try to translate a phrase that is not covered by the training data, or only has very few occurrences in the training data, the system can rarely succeed in producing a good translation, even if the translation is very easy from a human’s view. Numbers, times and dates are such examples. A language learner can master how to translate an arbitrary number in a small amount of time, but a statistical machine translation can hardly do a good job on this task even if it is trained with millions of data samples. The reason is that, however large the training corpus is, it can never cover all of the possible numbers, times or dates. The results coming out of the statistical system are often remarkably wrong for these kinds of translations, as illustrated in Figure 4-7.

Input: eight singles and eight quarters, please.
Output: 一美元 和 八 八个 两 角 五分 的 辅币。
(one dollar and eighty eight quarters)

Input: my watch loses ten minutes a day.
Output: 我 的 表 一 天 慢 三 十 分钟。
(my watch loses thirty minutes a day)

Figure 4-7 Problematic translation of numbers for statistical machine translation system

In the first example, the number "eight" is doubled and becomes eighty eight. In the second example, "ten" is translated into "thirty". It is obvious that the two translation are not acceptable even though the sentence structures are correct. In the travel domain, many sentences contains numbers, dates or times. They may involve a monetary amount, the time of an appointment, the date of the flight, etc. The information carried by numbers is so important that, if they are translated wrong, it will cause severe misunderstanding. Thus the translation needs to be precisely correct, and it is not a difficult task to translate them linguistically. We can pre-translate them directly into target Chinese, and replace them with a unique tag before feeding the sentences into the statistical system.

The pre-translation is done directly by GENESIS, as part of the Zhonglish generation. GENESIS not only can generate a string according to the rule templates, it can also map words according to a context-dependent lexicon. When generating numbers, times and dates from the semantic frame, we use the lexicon to map the English words into Chinese characters, and put HTML-like tags around the pre-translated phrases, for example <$number>, </$number>, etc. Then a simple script takes out the Chinese translations and stores them in a separate file for later access, and leaves the sentences containing only the abstract tags. The script also numbers the tags in case there are two tags of the same kind in one sentence.

The tagging process should be performed on the parallel Chinese training sentences too. To tag the Chinese sentences, we use a fragment parser instead of a full parser. The fragment parser doesn't try to parse a sentence into one parse tree. It only looks for phrases that satisfy a certain grammar. The grammar is typically small. For example, here, the grammar only covers numbers, dates and times. Figure 4-8 shows a snippet of the grammar. The fragment parser starts from the beginning of the sentence, and tries to parse as many words as it can. If it succeeds, it will tag the phrase according to the instructions from a special grammar category. Otherwise, it skips the first word, and starts again from the second word. The fragment parser is faster than the standard full parser, and disregards how complex the sentence is: it can perform well tagging the pieces we want.

```
.Sentence
a_number
when_node
....

.a_number
Numeral
[zero] digits
....

.*tagged_categories*
a_number
clock_time
month_name
...
```

Figure 4-8 A snippet of the fragment grammar

After the sentences are tagged, we generate a language model using the abstracted Chinese sentences. The abstracted Zhonglish and Chinese pairs are used to train the statistical machines translation system, so the trained model will not deal with any concrete numbers, dates or times. When testing, we post-processed the outputs produced by the statistical system. The unique tags are restored with their actual Chinese word strings. Table 4-2 demonstrates the whole procedure.

English Input	eight singles and eight quarters please .
Zhonglish	please <$number> 八 </$number> singles and <$number> 八 </$number> quarters period*
Abstracting	Abstracting: please <$number1> singles and <$number2> quarters period*
Output of SMT	请 给 我 <$number1> 张 <yi1> 美元 和 <$number2> 个 <aquarter1> 的 辅 币 。
Final Translation	请给我八张一美元和八个两角五分的辅币。 (Literally: Please give me eight one dollars and eight quarters.)

Table 4-2 Process with pre-translation

For the experiments in this chapter, we tag and pre-translate the following phrases:

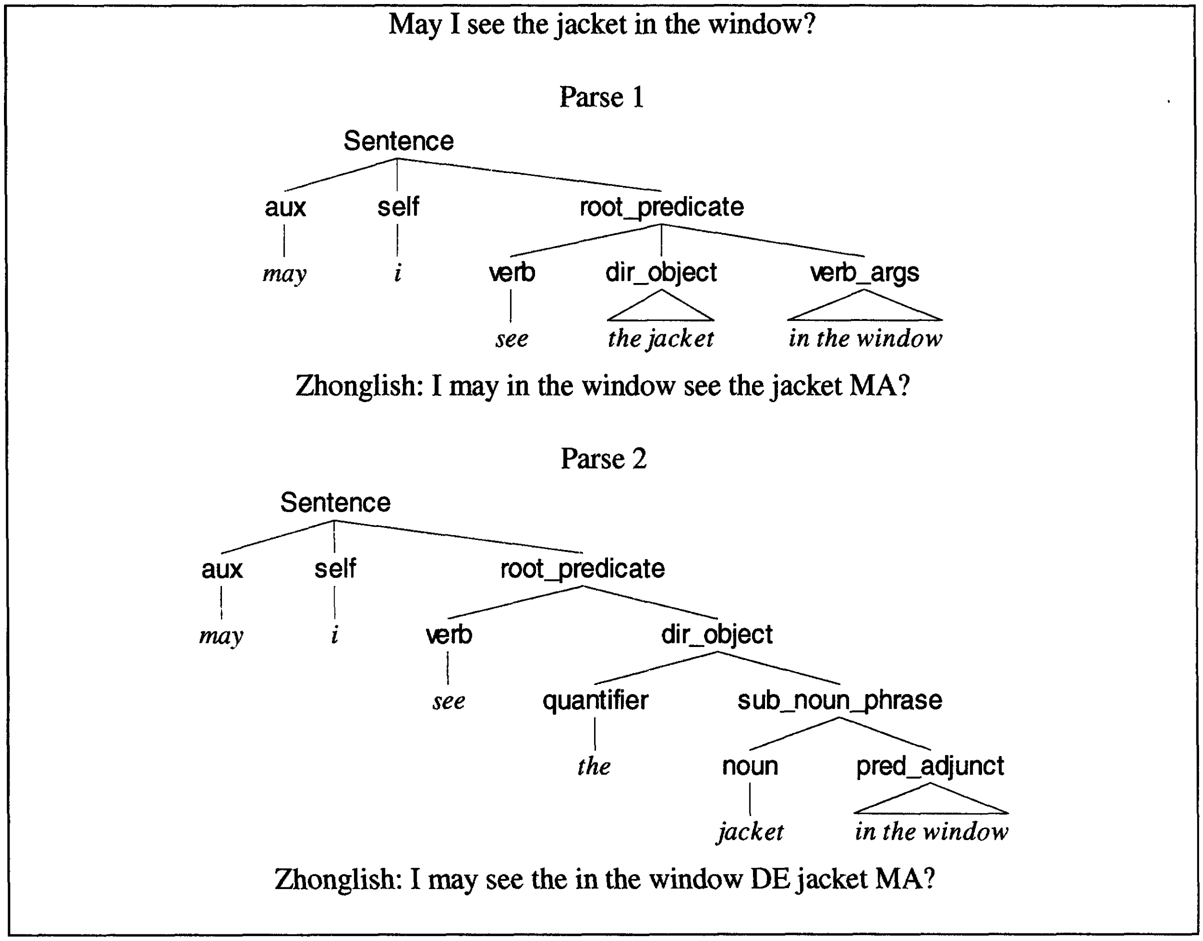
- Clock time: four fifteen, half past seven, eight o'clock a.m, etc.
- Month name: January, February, etc.
- Day number: the number after the month name.
- Weekday: Monday, Tuesday, etc.
- Year: two thousands and one, etc.
- Number: seventy four, two three three one, etc.
- N-th: first, eighth, etc.

And we tag "one" as a special category <yi> on the Chinese side, since the Chinese "one" sometimes corresponds to a number in English, but at other times it is just the determinative "a". Although we mainly deal with phrases related to numbers in the experiments, the idea can be pushed forward to pre-translate other linguistically confident pieces, and gradually lessen the work of the statistical system.

4.4 Disambiguating Prepositional Phrase Attachment

Another problem we met with is the ambiguous parse. Although the generation rules we write reflect the grammar of Chinese as much as possible, it is not always showing the correct

Zhonglish output. The reason is that GENESIS gets the incorrect parse among all the other alternatives from TINA. The typical example is the well known classic PP-attachment problem. The generation rule is simple enough to prepose the prepositional phrase ahead of verb or noun, but the correctness of the Zhonglish output depends on the correctness of the parsing. If the PP is incorrectly attached, it will show up as a gross error in the Zhonglish translation, and very likely affects the final Chinese translation too.



May I see the jacket in the window?

Parse 1

Zhonglish: I may in the window see the jacket MA?

Parse 2

Zhonglish: I may see the in the window DE jacket MA?

Figure 4-9 Ambiguous parses of a sentence exhibiting the PP-attachment problem. The parse trees are simplified to show the overall structures.

An example is given in Figure 4-9. The sentence “May I see the jacket in the window?” has two ambiguous parses. The prepositional phrase “in the window” can be attached to either the verb “see” or the noun “jacket.” The two different attachments result in two different Zhonglish outputs. "In the window" ends up landing right before the word it is attached to. Also, when the prepositional phrase is modifying a noun phrase, an additional nominal auxiliary word "DE" is necessary. In order to generate a correct Zhonglish sentence, which in this case is the second choice, we need a way to choose the right parse.

One way to solve this problem, of course, is to improve the parser, to have the parser put the correct parse on the top choice. Although TINA does have a probability model that can bias it towards the correct answer, the probability model is not lexicalized. For many cases of the PP-attachment problem, it is the combination of the verb, object and preposition that decides where the prepositional phrase should be attached, i.e. which word is the prepositional phrase more likely to modify. We need to look for a way that can automatically collect information about the likelihood that this verb/noun can be modified by this prepositional phrase. We were inspired by the difference between the two Zhonglish sentences in Figure 4-9. The explicit striking differences in the surface forms of Zhonglish imply that we can use statistical language modeling techniques applied to the Zhonglish string to potentially reduce errors.

The idea is implemented as follows. In order to use a language model to score the candidates, we first need a set of Zhonglish sentences that contains correct PP-attachment information to generate the language model. These sentences are obtained by GENESIS generation using a set of so-called "conservative PP-reordering rules". In this set of generation rules, only those prepositional phrases that are not ambiguous are reordered to the preposed position. The situations where the prepositional phrases are not ambiguous include:

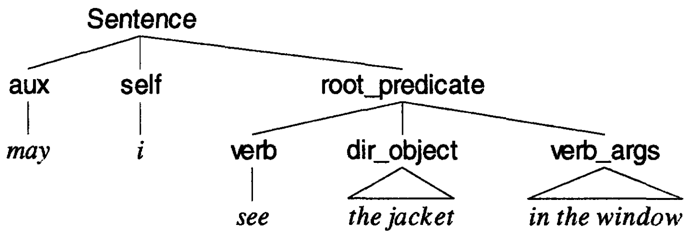
- Prepositional phrases inside the subject noun phrase
- Prepositional phrases inside an intransitive verb phrase
- Prepositional phrases inside a transitive verb phrase, when the object of the verb is a pronoun

Other possibly ambiguous prepositional phrases, typically in the situation of VP NP PP, are still left behind as if expressed in English. In this way, the Zhonglish sentences generated are guaranteed to have all preposed prepositional phrases in correct positions. We train an *n*-gram language model on these conservative PP-reordering Zhonglish sentences.

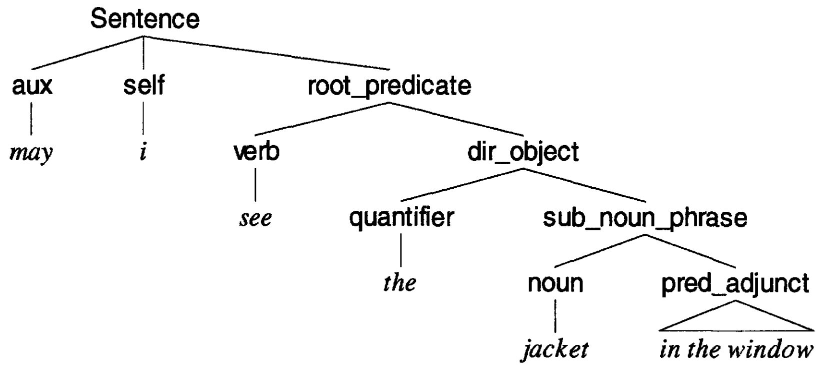
Then we run Zhonglish generation for a second time. This time, the ordinary Zhonglish generation rules are used, in which, regardless of possible ambiguity, all the prepositional phrases are reordered into the preposed positions. We configure TINA to let it output N-best semantic frames from N-best parses. Successively, GENESIS generates N-best Zhonglish sentences for these N-best semantic frames. Finally, the n-gram language model, which is trained using the above procedure, scores the N-best Zhonglish sentences and selects the best-scoring one.

The trick happening here is that, although in the language model there are many instances where the prepositional phrases are post-posed, these statistics are not used. The ordinary Zhonglish generation rules force all the prepositional phrases to reorder. They cannot stay at the end of the sentence except in some cases where they are the complement of the verb rather than an adjunct. So only the *n*-grams relating to the pre-posed prepositional phrases will take effect. They contain information about whether the prepositional phrases prefer to be attached to the specific verb or to the specific noun. And this information is all correct since it is from unambiguous sentences. Thus, compared to the raw result coming out of the parser, in which the PP-attachment is nearly random, the one chosen by the *n*-gram language model will have a much greater likelihood to be correct. Figure 4-10 shows some examples of results before and after this process. For the first example, the parser initially decides to attach the prepositional phrase to the verb, which is wrong. After the scoring and re-ranking, the correct attachment to the noun is chosen. On the other hand, for the second example, the initial decision is to attach the prepositional phrase to the noun, but is wrong this time. And, after the process, it is correctly attached to the verb.

May I see the jacket in the window?

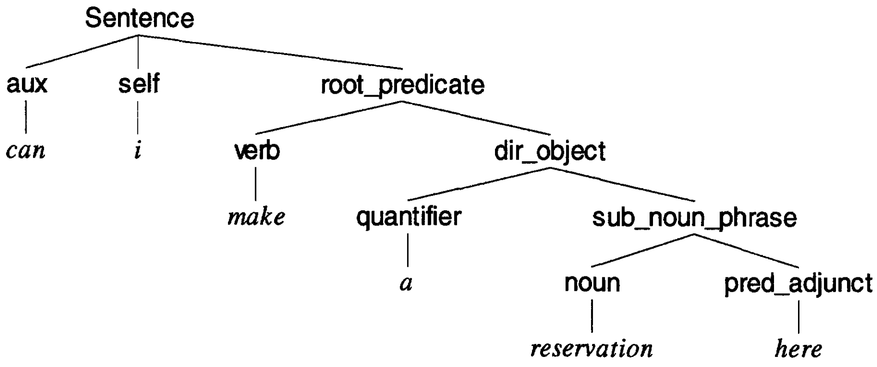


*Zhonglish: I may in the window see the jacket MA?



Zhonglish: I may see the in the window DE jacket MA?

Can I make a reservation here?



*Zhonglish: I can make a here DE reservation MA?

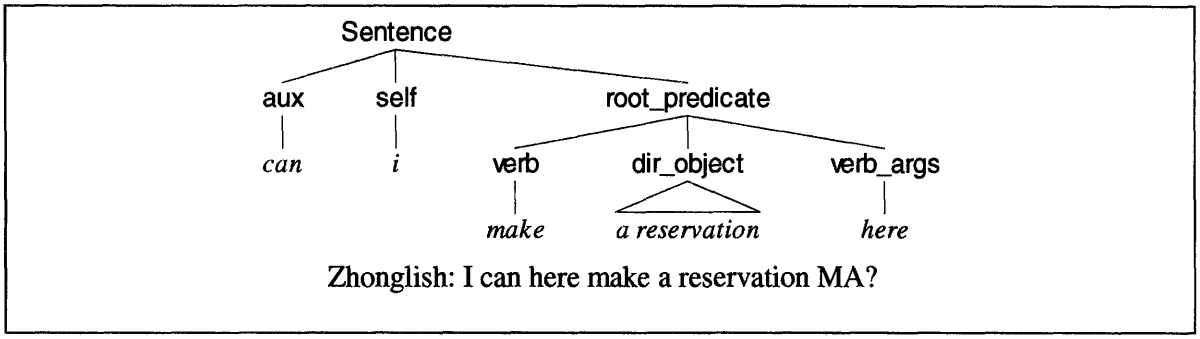


Figure 4-10 Two example of best parses and their Zhonglish paraphrases before and after PP-attachment disambiguation process. The parse trees are simplified and the (*) indicates the wrong Zhonglish.

4.5 Brief Summary: System Framework

In the last three sections, several problems were discussed. Figure 4-11 and Figure 4-12 summarize the translation system framework for training and testing that integrates all the methods talked about above.

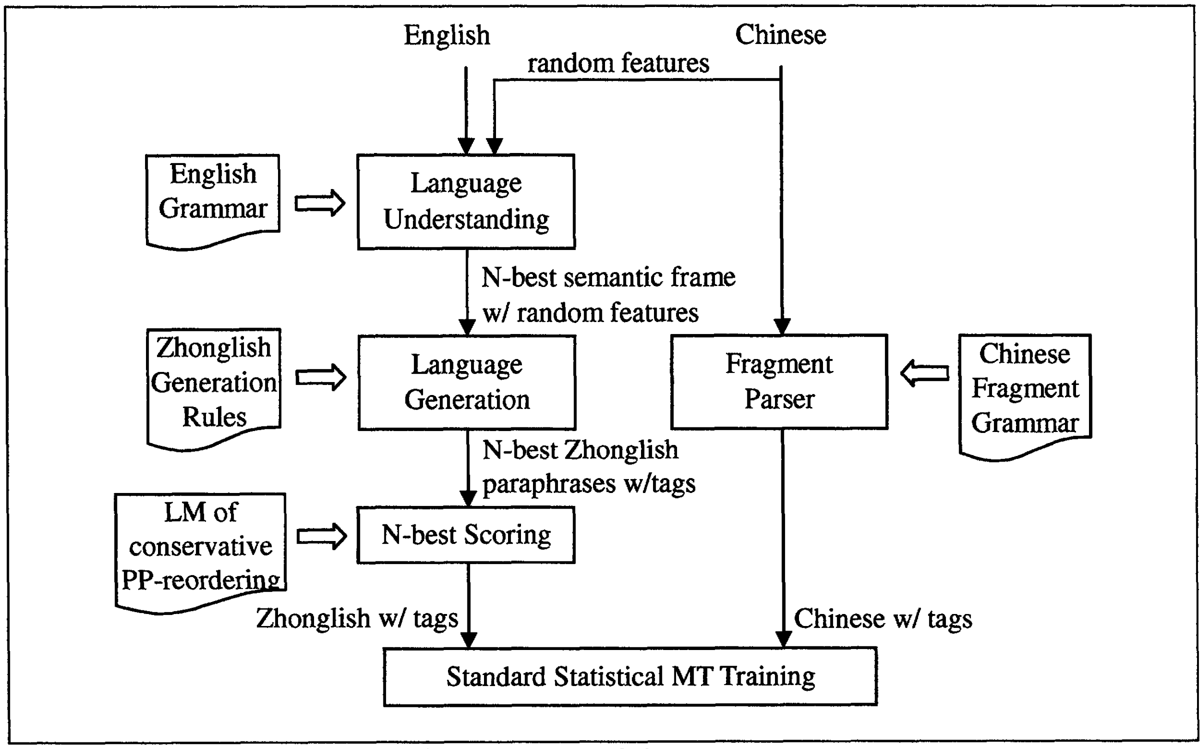


Figure 4-11 Translation system framework: training

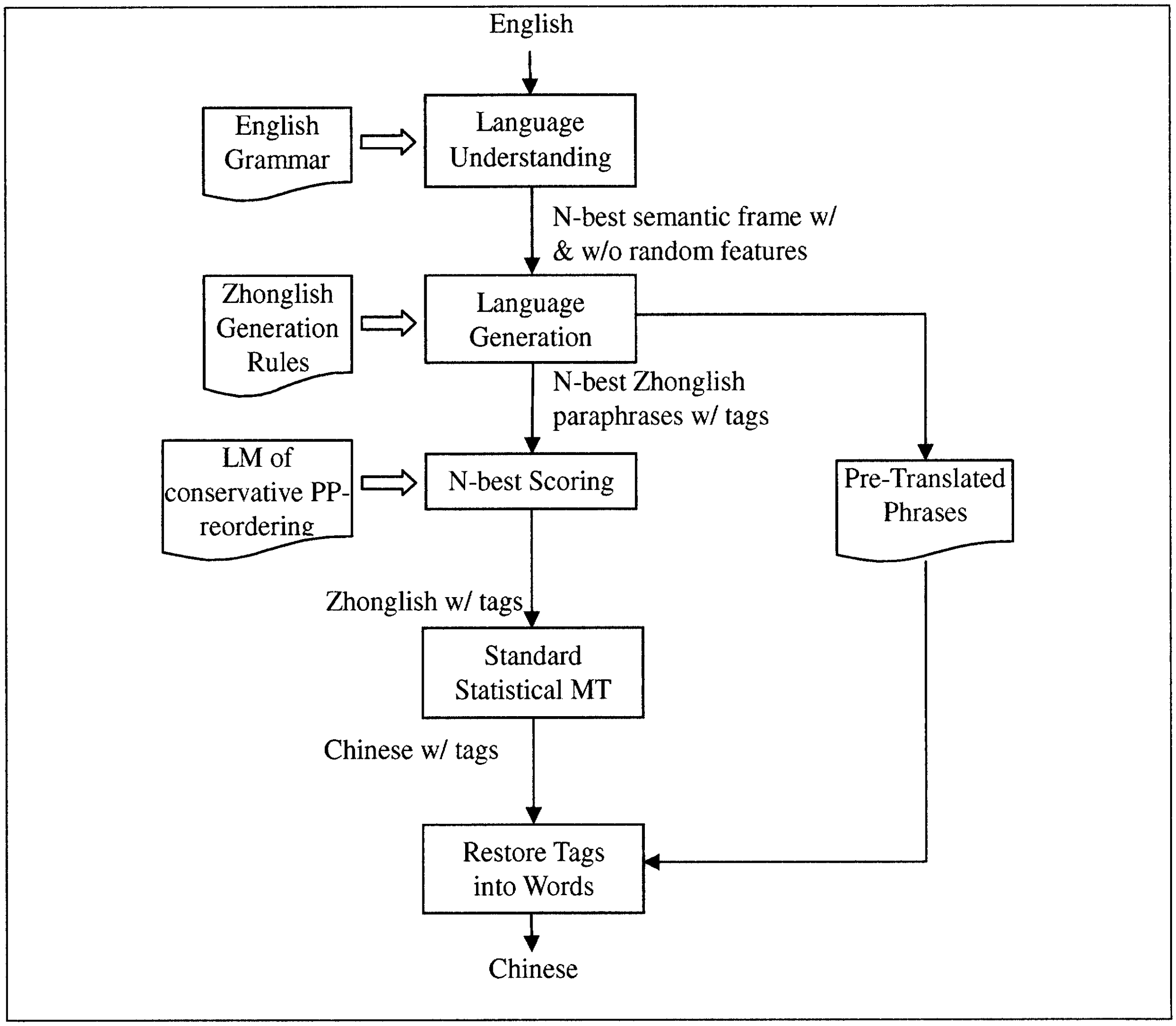


Figure 4-12 Translation system framework: testing

4.6 Experiments and Results

4.6.1 Experimental Setup

The statistical machine translation system we use is the phrase-based statistical machine translation system MOSES [Koehn, et al, 2007]. This also serves as the baseline. The maximum reordering distance parameter is set to six both for the baseline and for working in our combined system. The corpus is the IWSLT-06 data, which is a domain of travel and tourist information. The training set consists of 39,952 parallel sentences, among which about 20% are wh-questions. We use two held-out development sets: dev-set 1 as our development set to do minimal error training, and dev-set 2 as our test set. Both sets have approximately 500 sentences. The experiment is English-Chinese translation.

4.6.2 Results

The baseline system is trained with the full training set. However, for our system, the language understanding system cannot parse all of the source sentences. We throw away the sentences that cannot be parsed or can only produce a partial parse result. This gives us a parsable set of 35,672 parallel sentences for training, which is about 89% of the full training set. The same thing is done to the development set. The BLEU scores reported in Table 4-3 are all tested on the parsable set of the test data, which consists of 453 sentences, 90.6% of the full test data set.

	Baseline (trained with full set)	Baseline (trained with parsable set)	Our Approach (trained with parsable set)
BLEU	31.48	30.78	33.33

Table 4-3 BLEU score of baseline and our system.

As shown in Table 4-3, even with 11% less training data, our approach realized a 1.85 point improvement on BLEU score over the baseline. When the baseline is restricted to train on only the parsable data, our approach gained over 2.5 BLEU points. Table 4-4 shows some comparisons of results from the baseline and our approach.

In analyzing the results, we concluded that the gain is mostly due to three reasons. First, proper reordering on the input string helps the output translation to be in the correct word order. The first example in Table 4-4 exhibits this point. The output from the baseline is almost a straight word-to-word translation. The meaning is quite different from the original English sentence. But with the reordering of our approach, the sentence is correctly translated. Secondly, pre-translation helps. Our system can always get the numbers, times and dates correct, as shown in the second example. Finally, the third example shows that, by reordering, the statistical MT system can better align the parallel sentences, thus producing a lexicon and phrase table with higher quality.

Input English	Results from Baseline System	Results from Combined System
what time does the bus for boston leave?	什么 时候 的 巴士 从 波士顿 出发 ? (The bus of what time leaves from Boston?)	这 趟 去 波士顿 的 巴士 什么 时候 出发 ? (When does this bus for Boston leave?)
that comes to five dollars and thirty-two cents.	总共 两 美元 三 十 五 美分 。 (Altogether two dollars thirty-five cents.)	总共 是 五 美元 三 十 二 美分 。 (Altogether is five dollars thirty-two cents.)
it's cherry blossom season.	这 是 cherry blossom 季节 。 (This is cherry blossom season.)	它 是 樱花 的 季节 。 (It is cherry blossom's season.)

Table 4-4 Examples of different outputs from the baseline and the combined system

4.6.3 GIZA Alignment Cross-Over

Despite the improvement we got for BLEU score, we observed that this is not a very reliable measurement for our experiment. We only have one reference translation for each test sentence, and our test set is so small that a tiny meaning-preserving change in the translation output can cause the overall BLEU score to fluctuate on quite large a scale. Furthermore, some reference sentences have typos and some are not judged as the natural way of expressing the meaning by native speakers. Therefore, we want to consider a more intuitive and more direct way of evaluation.

Since our purpose was to introduce an intermediate language, so that the parallel sentences fed into the statistical system are alike in terms of word order, it becomes natural to think of directly measuring how alike the Zhonglish sentence and actual Chinese sentence are. The method we adopt is to calculate the crossover of their GIZA-alignments coming out from training.

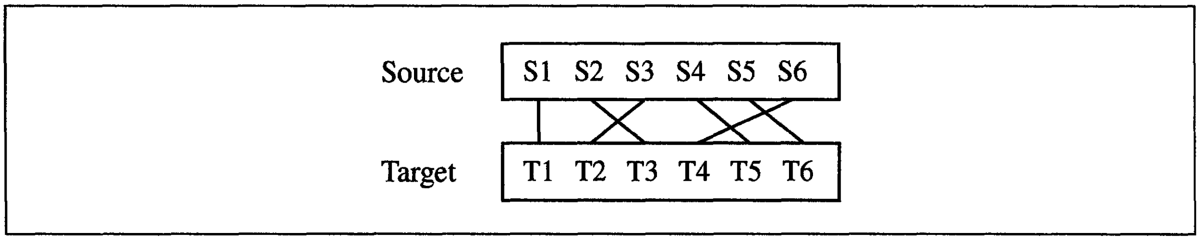


Figure 4-13 Example of calculating GIZA-alignment crossover. In this example, the crossover equals to 3

Figure 4-13 exemplifies how the crossover is counted. Each cross is counted as one. If an alignment crosses two other alignments, it is counted as two crossovers. This means long distance disorder is worse than local disorder, which conforms to how the statistical machine translation system penalizes reordering.

This measurement is objective and easy to obtain. Ideally, if our Zhonglish is perfect, all the alignments should be straight. There would be no crossovers. Although the ideal case can never be achieved, it is very intuitive from the number to see how well our reordering is doing. Table 4-5 lists the average crossover of the training data of the baseline system (pair of English and Chinese sentences) and our system (pair of Zhonglish and Chinese sentences). Our Zhonglish-Chinese pair has only about 55% as many crossovers as the original English-Chinese pair. And we have 18% more sentences that are crossover-free. This is a substantial improvement. We also measure how our dealing with random constructions and ambiguous parses affect the crossover. Table 4-6 gives the result.

	Average Cross-over Per Sentence	Normalized Average Cross-over	% of Zero-Cross-Over Sentences
Eng-Chn Pair (full training set)	11.3	0.97	22.0
Eng-Chn Pair (parsable training set)	10.52	0.95	22.5
Zhn-Chn Pair (parsable training set)	5.91	0.52	40.5

Table 4-5 GIZA-Alignment Crossovers for original English-Chinese Pair and our Zhonglish-Chinese Pair. The normalized average crossover is obtained by normalizing on the length of English/Zhonglish sentences, since they're not necessarily the same length.

The figures show positive results of our methods for handling the two problems. Especially for the handling of ambiguous parses, the decrease in crossover indicates that more prepositional phrases are in the correct position; i.e., they are attached to the correct phrases.

	Average Cross-over Per Sentence	Normalized Average Cross-over	% of Zero-Cross-Over Sentences
Zhg-Chn Pair	6.08	0.54	39.4
Zhg-Chn Pair (+RC)	6.03	0.53	40.0
Zhg-Chn Pair (+RC, +AP)	5.91	0.52	40.5

Table 4-6 Effects of random construction and ambiguous parses handling. RC stands for Random Construction. AP stands for Ambiguous Parses.

Chapter 5. Translation Game for Chinese Learning

The last two chapters discussed two technologies: parsing Chinese in an efficient way and translating English into Mandarin Chinese using a combined linguistic-statistical method. Both technologies showed positive results in stand-alone experiments. With these two technologies, it is now feasible to build a translation game system for learning Chinese in a broader domain. Although this thesis does not actually build the Web-based system, the core technology components of the user interface are developed. In this chapter, I will talk about how the two technologies described in the previous chapters are adapted into such a language learning system, and how well they perform in the integrated system.

5.1 Framework

Figure 5-1 shows the framework of the translation game system, which is leveraged from the two previous language learning systems. The figure excludes the speech recognition component and assumes the translation from the student is obtained in a text format, which is not true in the real case. The system works as follows. The system first generates an English sentence for the student to translate according to the current difficulty level. After the system has received the translation from the student, the translation is sent into the language understanding component, namely TINA, fed with a set of Mandarin Chinese grammar rules. If the sentence is successfully parsed, the language understanding component produces a test semantic frame, which will be compared against the reference semantic frame. In the mean time, the original English sentence goes through the English-Chinese machine translation component. A reference Chinese translation is produced and sent into the same language understanding component. The outcome will be the reference semantic frame. Then the meaning comparison will be carried out between the reference semantic frame and the test semantic frame. If the comparison is positive, the system congratulates the student and prompts for the next sentence; otherwise, it tells the student to try again. At any time, the reference translation is available to the student by clicking a button on the user interface.

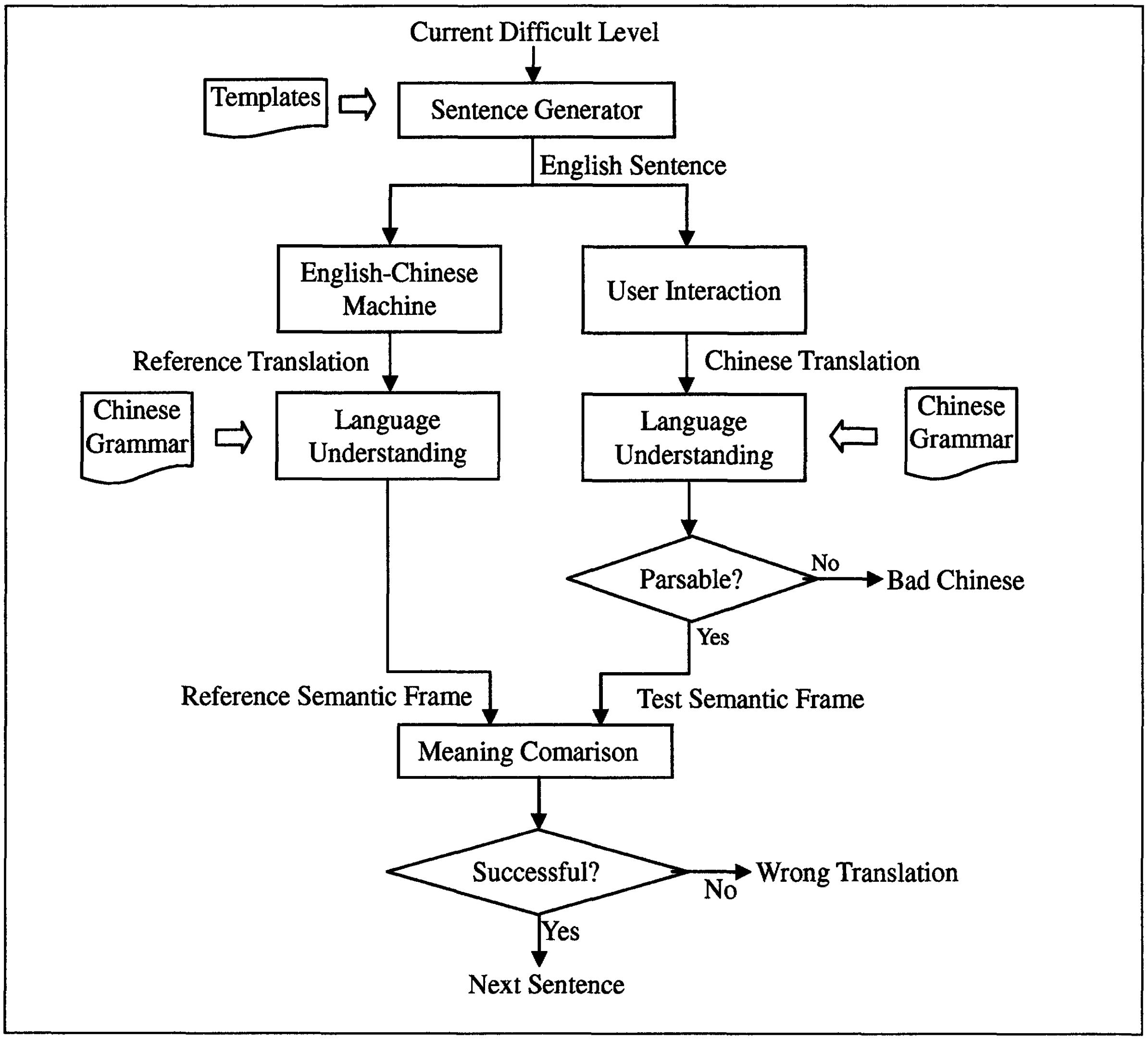


Figure 5-1 Framework of the System (Excluding Speech Recognition Component)

The framework is very much the same as in the previous two language learning systems, but it is crucially different in some ways. First, the domain this framework will apply to is the travel domain, which means the system is going to have richer templates for generating English sentences, more sophisticated Chinese grammar to handle parsing and more complexity in meaning comparison. Second, the language understanding component is the TINA system enhanced with restructure and other features to help parsing Chinese better and faster. Third, the English-Chinese machine translation component is the combined system of both linguistic and statistical methods.

Although it may seem straightforward to adopt the technologies directly in this framework of translation game, it turns out that more investigation is needed in order to make the system work in an acceptable manner. In the following sections, I will discuss these issues in detail.

5.2 Generating English sentences

Giving the students appropriate sentences to translate is important in any translation-based language teaching method. In traditional classrooms, teachers and textbooks typically prepare the sentences that can help the students master important grammar structures and new vocabulary. We want to have our system provide a similar ability. Since our system is solely a translation game system without additional text or grammar explanation, we would like to start from really simple sentences so that the students can easily figure out the Chinese syntax and words with the help of the reference translation. Then, gradually new structures and vocabulary are introduced, so the students will learn more and more. We also want our system to provide a very large number of sentences, so that the students will not always be asked to translate the same sentences over and over at a certain level.

To implement these two ideas, we first carried out a selection procedure on the original IWSLT corpus. All the sentences were broken down into short pieces, which contain at most one period, question mark or exclamation mark. Then we calculated the average word unigram score for each of the short sentences. Because the unigram score reflects how frequently a word appears, a sentence with low average unigram score means the words it contains are on average frequent, or common. Those sentences with difficult or rare words can be expected to have high average unigram scores. We sorted the sentences by the average unigram score, and only select the sentences that contains the top 1,000 most frequent Chinese words. These sentences, together with their English parallel sentences, gave us a set of around 15,000 sentence pairs.

We looked at these 15,000 sentence pairs, and found that many Chinese sentences have multiple occurrences, and each is translated into different English. This happens to the English side, too. Although these translations are all correct, we don't want to confuse the beginner student, or to have them remember all these various ways to translate. So we further eliminated the multiple translations, and only kept the one with the lowest average unigram score. At this point, the set was reduced to about 12,000 sentence pairs.

Using these 12,000 sentence pairs as reference, we built lesson templates. Each lesson focuses on one topic, and has several new sentence patterns with associated new vocabulary. It can also use the patterns and vocabulary introduced in previous lessons. A program developed in our group can easily read in the templates and generate well-formed sentences. The templates are hand coded, but it is not too bad to do so. In fact, this is just like composing an ordinary Chinese textbook, but with the computer, it can generate many more sentences for the student to practice than any paper text books.

5.3 Generating Reference Translation: Zhonglish II

In Chapter 4, I showed how the linguistics-based parse-and-paraphrase system and statistical phrase-based machine translation system can be combined to perform English-Chinese translation. The result of the combined system outperforms the pure statistical system in terms of both BLEU score and GIZA alignment cross-overs. Though the improvement of the numbers is encouraging, it is still not satisfying for the game system. In language learning systems, errors in reference translations are not tolerable. They will confuse and mislead the students. What we want is absolutely correct translations, which under current technology is an impossible task in general. However, the situation we face is a little different. We do not hope to translate an arbitrary English sentence perfectly into Chinese. The demand is only within the domain that the system will use to teach the students. Furthermore, since we aim at beginners, difficult and rare vocabulary items are avoided intentionally. More specifically, as described in the last section, we constrain the vocabulary to 1000 most common Chinese words. So the task becomes high-quality English-Chinese translation within the travel domain and with limited vocabulary, which is not impossible anymore.

To achieve the goal, the basic paradigm is still the combined translation system. But this time, we would like to put more emphasis on the linguistic part. There are two reasons for doing this. First, linguistic generation is more controllable than the statistical translation system. We can easily have a sentence translated into this way or that by altering the rules, but it is really hard to intervene in the statistical system to let it produce the result we have in mind. Second, our sentences are generated from templates. This means the system will need to translate a lot of sentences that don't show up in the training data. Even if the sentence is only different by a synonym noun, it is a totally new sentence to the statistical system. A phrase-based translation system trained with only tens of thousands of sentences cannot be expected to generalize well enough on these new sentences. However, for the linguistic system, it is simply a lexicon substitution with all the other structures kept the same.

As a result, we would like to push forward the intermediate language "Zhonglish" and make it more like Chinese. We call it "Zhonglish II" to distinguish it from Zhonglish, in which most of the words are still in English. In Zhonglish II, Chinese characters appear and more intricate grammar structures are taken care of. In the following sub-sections, I will describe how we evolve Zhonglish into Zhonglish II with the help of the statistical machine translation system.

5.3.1 Semi-Automatic Lexicon Acquisition

When we decided to let Zhonglish II contain actual Chinese characters, it means we need a lexicon that maps English words into Chinese. Although we have constrained the vocabulary into the 1000 most common words on the Chinese side, which corresponds to about 2100 unique words for the English side, it is still a daunting job if the lexicon is going to be created by hand. Also, the translation will likely be inaccurate without actually looking at the context. Thus, a more automatic machinery which can extract a lexicon from the parallel corpus is desired. And this is not hard to realize, because it is exactly one of the steps in the statistical machine translation system.

Usually, the way that a statistical machine translation system extracts the lexicon is by performing GIZA alignment. The aligning makes use of techniques from machine learning and tries to predict how the words in the source and target sentences can be aligned. After the alignment is produced, the corresponding pairs of words form the lexicon for the statistical machine translation system.

The process is very close to what we want, but we need some extra modification. The GENESIS system is a linguistic-based system. It honors part-of-speech information since oftentimes a word has different translations according to different part-of-speech categories. So we want to automatically obtain the part-of-speech information as well as the English-Chinese word pair. Another important thing is that the lexicon produced by statistical methods is usually very noisy. We need to clean it up by some automatic and/or manual methods.

Here is our solution. We use an iterative process to gradually obtain a larger and larger lexicon with a human in the loop. The lexicon entries are produced mainly by automatic machinery, and the role of the human is to refine and delete the wrong entries. The following paragraphs describe the iteration steps in more detail.

Step 1, Tag each word of the Chinese sentences. This is done by TINA. TINA parses the sentences and appends the name of the parent node to each terminal word as a class tag. The class tag is appended to the word without spaces, so from the view of GIZA it is one word. In principle, it would be better to tag the source side of the parallel corpus for part-of-speech information. But since Zhonglish and Zhonglish II are man-made intermediate languages with a mixture of English and Chinese words, it is not easy to write a grammar to parse the sentences. So we only count on the Chinese side for the part-of-speech information.

Step 2, Align Zhonglish (or Zhonglish II) with the tagged Chinese. We use GIZA++ to do the aligning. The system takes in the sentences and produces an alignment file.

Step 3, Post-process the alignment file and obtain a raw lexicon. Aligned word pairs are retrieved by using the alignment file. The class tags in the Chinese words are separated out and the part-of-speech information is used to group the word pairs into categories. We prune away the function word categories and only keep the noun, verb, adjective and adverb categories. The translations of function words like prepositions, auxiliaries, etc. tends to be more variable and inconsistent. GIZA usually doesn't do a good job in finding alignments for these words. And there are only a few words in these categories, so we would rather prepare the lexicon entries of these words manually.

Step 4, Refine the raw lexicon. The raw lexicon produced by GIZA is noisy. Each word typically has more than one translation. For common words, there may even be over ten or twenty translations. Not all the translations are correct, of course; indeed, only several among all the possible translations GIZA gives out are correct. To obtain a cleaner lexicon, we first count how many times each translation occurs according to GIZA's alignment output file. Those with few occurrences compared to the other candidate translations are pruned. Then we have a human in the loop to look at the rest of the lexicon, and manually decide which entries are going to be kept. This may seem to be an overwhelming job for a human to look through all the lexicon, but it is not as hard or time-consuming as it seems to be. Our vocabulary is constrained down to 2100 English words, and, as mentioned above, about two-thirds of the entries in the lexicon can be easily determined as junk. The decision is quite objective for a bilingual speaker. Usually, it costs less than one hour to go through the raw lexicon of over 4000 entries and produce a clean version.

Step 5, Merge the lexicon into GENESIS. This step is easy. We copy all the entries in the clean lexicon into GENESIS's vocabulary file. The only trick is for the words that have multiple translations with different semantics, only the most common one is picked at the beginning. Later on, other translations are gradually added using the selector feature of GENESIS to indicate the context information.

Step 6, Produce the new version of Zhonglish II sentences and start over from Step 2. With the new lexicon, Zhonglish becomes Zhonglish II, which contains actual Chinese words. This means the output now looks more like the final Chinese translation. So, if we iterate the lexicon acquisition process, we will get better alignment, and very likely we can obtain new lexical entries that didn't appear or were pruned away due to low confidence in the last iteration. The new entries again are merged into the GENESIS vocabulary file, and a second new version of Zhonglish II can be generated.

Iterating the above six steps, a high-quality English-Chinese lexicon with part-of-speech and some context information can be obtained without much manual effort. We iterated the process four times, and obtained a lexicon of about 1265 entries of nouns, verbs, adjectives and adverbs. Every iteration produces new lexicon entries. Table 5-1 shows the rough numbers of new entries obtained in each iteration. The figures are not accurate because there are duplicated instances between the newly obtained entries and existing entries, and there are other manipulations performed on the GENESIS vocabulary file between two iterations. Together with the other manually written entries, which mostly focus on function words, we finally obtained a lexicon of about 1650 entries.

# Iteration	Number of new entries obtained
1	886
2	259
3	75
4	60

Table 5-1 Number of new lexicon entries obtained in each iteration

5.3.2 Promoted Predicate and Frame Transformation

One problem we encountered when producing the lexicon concerns English verb phrases. English has a large number of verb phrases which are composed of a verb and either a preposition or an adverb particle. For example "think about" is a phrase consisting of the verb "think" and a preposition "about". On the other hand, "try on" is a phrase consisting of the verb "try" and particle "on". "On" is not a preposition in this case because "try on it" is ungrammatical. The meaning of such a phrase is usually the integration of the verb and the preposition/particle, instead of simply adding the meaning of the two together. When translated into Chinese, the phrase corresponds to one Chinese word. For example, "think about" is translated as "考虑" instead of literally "觉得 关于". So we need a way to tell GENESIS to generate "考虑" when "think" is accompanied by "about", and generate nothing for the "about".

The usual way to do this is by setting GENESIS flags. The preposition "about" sets a flag, and when GENESIS sees this flag, it will generate a special translation for "think". Similarly, "think" sets another flag to tell GENESIS to generate nothing for "about". But this method is quite annoying when there are so many verb-preposition/particle combinations of this kind. It would be good if, in the semantic frame, the name of the predicate is "think about" instead of "think" with another predicate "about" under it, so that we can easily put "think about" into the lexicon as one entry.

The proposal that comes up naturally is to register "think about" as one verb in the parser's lexicon. However, this is not a good solution. "Think" and "about" don't always stay together. They can be separated by an adverb, for example "think carefully about it". So trying to parse them as one word doesn't always work. We have to do some process to the semantic frame instead of the parse tree. And this process is done by two new mechanisms implemented in TINA, which can deal with verb-preposition combination and verb-particle combination respectively.

The first mechanism is called “promoted predicates”. This is an existing mechanism actually. It can promote one predicate to the parent level and optionally rename it. But what we want is a little bit different. The child predicate, i.e. the preposition, needs to be promoted and unified with the parent verb predicate. So a new feature “rename and unite” is added into the promoted predicates framework to allow the unification. The syntax to indicate the actions is simple, as shown in Figure 5-2. It says to promote the predicate “about” under the predicate “think”, rename it to “think_about”, make this new predicate inherit all the keys that are under “think”, and finally delete the predicate “think”.

```
{c rewrite_rules
        ...
        :promoted_predicates (
                {p think :promote_and_unite ( “about” “think_about” ) }
                {p ... } ) }
```

Figure 5-2 Syntax for promoted predicates

The promoted predicates can successfully solve the verb-preposition combination. However, the situation for verb-particle combination is different. In the semantic frame, the particle is not a predicate. It is a key with string value. To solve this problem, we use the frame transformation mechanism. The frame transformation mechanism was originally created to lessen the difference between semantic frames generated from different languages. It has a more complex syntax than the promoted predicates, and as a result, it is more powerful. Figure 5-3 demonstrates the frame transformation rules that can alter the frame for “try it on” into the desired form. The two rules work sequentially. The first rule looks for a predicate named “try” which has a :particle key with value “on”, and renames the predicate into “try_on”. Then the second rule deletes the :particle key from the predicate try_on. A full description of the syntax of frame transformation rules is shown in Appendix A.

```
{c rewrite_rules
        ...
        :trasformation_rules (
                {c transformation_rule
                        :in {p try :particle "on"}
                        :replace "*SELF*"
                        :with "try_on" }
                {c transformation_rule
                        :in {p try_on :particle "on"}
                        :replace "*SELF*"
                        :with {p *SELF* :particle "*NONE*" } }
                {c ... } ) }
```

Figure 5-3 Frame transformation rules to deal with verb-particle combination

The frame transformation rules can further deal with other troublesome situations, like verb-adjective phrase "get lost", verb-adverb phrase "get back", and so on. Figure 5-4 contrasts the original semantic frames with the ones that have been processed by promoted predicates and/or frame transformation. With the processed frames, now it is much easier to generate corresponding Zhonglish sentences.

Original semantic frames	Semantic frames after process
{c crequest :domain "LanguageLesson" :input_sentence "think about it" :pred {p think :verb* "think" :pred {p about :topic {q pronoun :name "it" } } } }	{c crequest :domain "LanguageLesson" :input_sentence "think about it" :pred {p think_about :topic {q pronoun :name "it" } :verb* "think" } }
{c crequest :domain "LanguageLesson" :input_sentence "try it on" :pred {p try :topic {q pronoun :name "it" } :particle "on" :verb* "try" } }	{c crequest :domain "LanguageLesson" :input_sentence "try it on" :pred {p try_on :topic {q pronoun :name "it" } :verb* "try" } }
{c crequest :domain "LanguageLesson" :input_sentence "get lost" :pred {p get	{c crequest :domain "LanguageLesson" :input_sentence "get lost" :pred {p get_lost

:verb* "get" :pred {p adj_complement :pred {p adjective :adj "lost" } } } }	:verb* "get" } }

Figure 5-4 Semantic frames before and after process

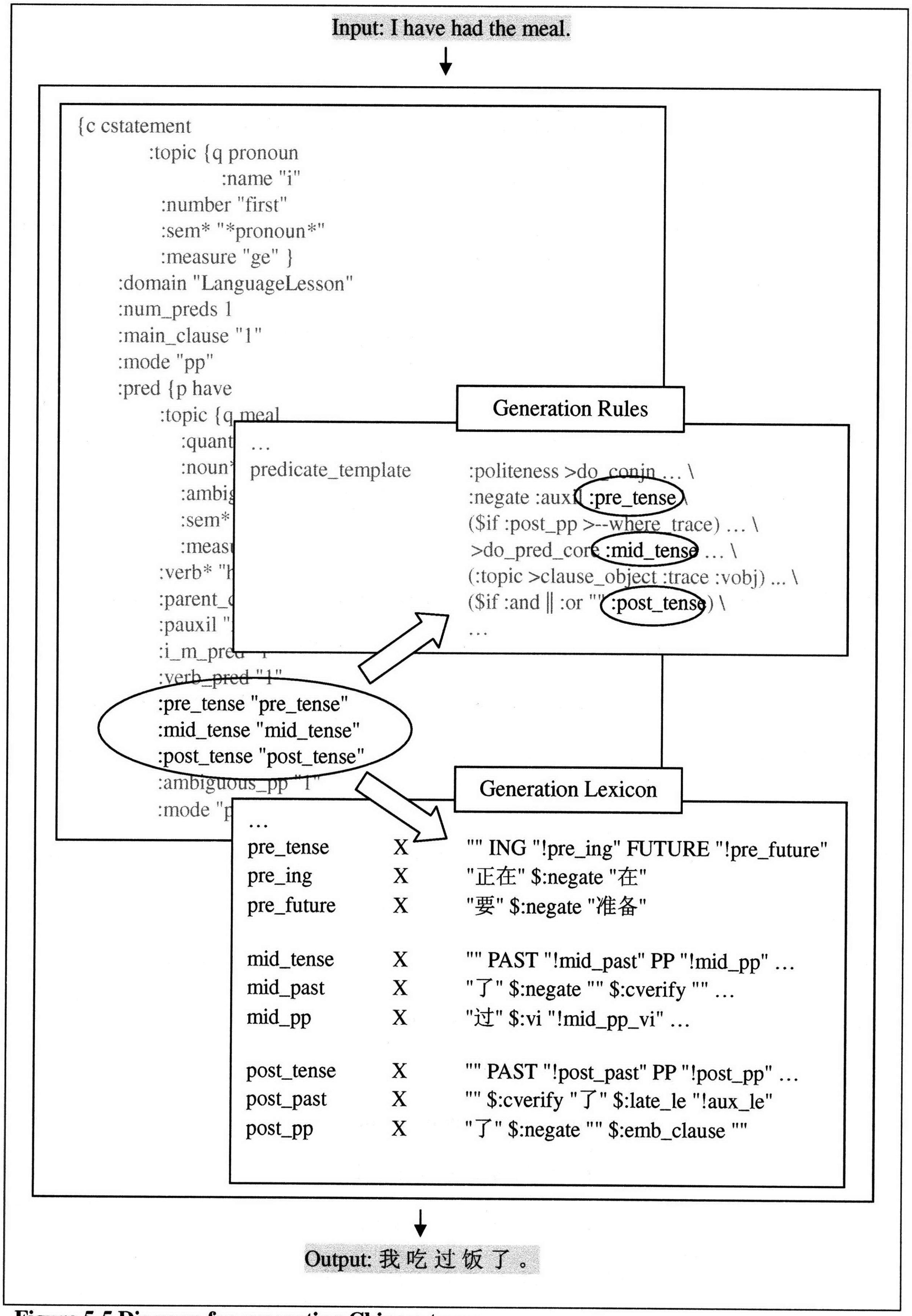
5.3.3 Generation Rules for Zhonglish II

The previous two sections talked about lexical issues for Zhonglish II. Not only did we make the words Chinese in Zhonglish II, but we also looked at structural issues. In the previous generation rules for the Zhonglish language, the main task is to arrange the constituents so that the sentence obeys the Chinese grammar on a high level. We dealt with wh-questions, prepositional phrase, adjective clauses, BA-constructions, etc. These are mostly high-level sentence structures, and we didn't pay much attention to the local disorder, in hope that the statistical system can help us with it. And indeed, it is hard to do more in Zhonglish, because Zhonglish words are mostly in English, and many ways of expression in Chinese don't have a literal correspondence in English. But now, Zhonglish II has most of the words in Chinese. It is time to devote attention to more precise rules to produce more accurate output, and further lessen the work of the statistical system. The rules are more intricate in many aspects. In the following several paragraphs, I describe some major improvements in Zhonglish II compared to Zhonglish.

Tense. Chinese has rather complex rules for tense, which we didn't focus on much in Zhonglish. But this time, we investigated those more closely. Chinese is a morphology-free language. Verbs in Chinese don't inflect to indicate tense. Instead, the tense is usually expressed by tense auxiliaries. There are three most common tense auxiliaries: "了"(LE), "过"(GUO), and "着"(ZHE). We specially focus on the first two, for ZHE rarely appears in a single-clause sentence.

There are two things that need considering for the Chinese tense auxiliaries. One thing is the position. Auxiliary GUO and ZHE only show up immediately after the verb, but LE is more flexible. It can appear right after the verb, or at the end of the whole verb phrase. The different positioning leads to slightly different emphasis for some verbs, and quite different meaning for other verbs. There are also cases where only one choice is grammatical. The other thing is of course, the meaning. LE roughly corresponds to past tense, and GUO roughly corresponds to perfect tense, but that is not the whole story. These two auxiliaries sometimes appear together in one sentence. When the verb is negated, LE disappears most of the time. When the sentence emphasizes locative, temporal or other adjuncts, for example a wh-question that asks the place where a certain action happened, the past tense is not characterized by the tense auxiliary LE, but by a tone auxiliary DE. And in embedded clauses like noun clause and adjective clause, many times the past tense is not explicit. With all these different situations, we want an easy and consistent way to generate the auxiliaries correctly.

The way we did this is that, unlike English, in which the tense causes the verb to be generated into different forms, we added explicit keys for tense auxiliaries into the verb predicate constituents in the preprocessor of GENESIS. There are three tense keys: :pre_tense, :mid_tense and :post_tense. :Pre_tense is the tense indicator located before the verb. It is created mostly for present continuous tense and future tense, where the indicators are actually adverbs instead of tense auxiliaries, but they function very similarly to the auxiliaries. :Mid_tense is the tense auxiliary immediately following the verb, and :post_tense is the one that is at the end of the verb phrase. In the semantic frames, the value for the three keys are the same as their key name. When generated, they are mapped into different words according to the tense and other flags. Figure 5-5 exemplifies the method.



Input: I have had the meal.

```
{c cstatement
        :topic {q pronoun
                :name "i"
          :number "first"
          :sem* "*pronoun*"
          :measure "ge" }
    :domain "LanguageLesson"
    :num_preds 1
    :main_clause "1"
    :mode "pp"
    :pred {p have
          :topic {q meal
            :quant
            :noun*
            :ambig
            :sem*
            :meas
          :verb* "h
          :parent_c
          :pauxil "
          :i_m_pred "1"
          :verb_pred "1"
          :pre_tense "pre_tense"
          :mid_tense "mid_tense"
          :post_tense "post_tense"
          :ambiguous_pp "1"
          :mode "p
```

Generation Rules

```
…
predicate_template        :politeness >do_conjn … \
                          :negate :auxil :pre_tense \
                          ($if :post_pp >--where_trace) … \
                          >do_pred_core :mid_tense … \
                          (:topic >clause_object :trace :vobj) ... \
                          ($if :and || :or "" :post_tense) \
                          …
```

Generation Lexicon

```
…
pre_tense      X      "" ING "!pre_ing" FUTURE "!pre_future"
pre_ing        X      "正在" $:negate "在"
pre_future     X      "要" $:negate "准备"

mid_tense      X      "" PAST "!mid_past" PP "!mid_pp" …
mid_past       X      "了" $:negate "" $:cverify "" …
mid_pp         X      "过" $:vi "!mid_pp_vi" …

post_tense     X      "" PAST "!post_past" PP "!post_pp" …
post_past      X      "" $:cverify "了" $:late_le "!aux_le"
post_pp        X      "了" $:negate "" $:emb_clause ""
```

Output: 我吃过饭了。

Figure 5-5 Diagram for generating Chinese tense

But that is still not the end of the story. The classification of tense in Chinese doesn't match that in English perfectly. For some English sentences expressed in the present tense, when translated into Chinese, the auxiliary LE is required, and sometimes vice versa. These kinds of mismatch have a lot to do with the property of the verbs. For example, in Chinese many opinion verbs like "like", "hate", etc. don't support the tense auxiliary LE. As a result, the translation for sentences like "I enjoyed the show" should be in present tense. To deal with this issue, we put additional tense conversion flags for the problematic verbs. For instance, the flag $:past_as_root tells GENESIS to treat the past tense as if it were present tense. There are not many of this kind of verbs in our limited domain, so it is doable.

GEI Construction. Both English and Chinese have a number of verbs that can take double objects. Most of the double object verbs in English when translated into Chinese are still double object verbs. But there are a few exceptions. For example in Figure 5-6, the translation for the word "show" is not a double object verb, so the indirect object "me" has to form a prepositional phrase using GEI. We call this GEI construction, because it is similar to the BA construction. The GEI construction is also a random construction to some extent, so we use the same technique as we did for BA construction.

English: Please show me your ticket
Chinese: 请 给我 看 你的 票子
Please GEI me show your ticket

Figure 5-6 Example of a GEI Construction

ZAI Construction. ZAI is the most common preposition in Chinese. Most of the prepositional phrases with structure "prep noun_phrase" become a form like "ZAI noun_phrase prep" when translated into Chinese. This kind of ZAI construction also suits adverbial clauses starting with a conjunction such as “when”, “before”, “after”, etc. While this is an easy rearrangement of word order, there is one noticeable issue. As mentioned before, in Chinese, some verbs take a locative prepositional phrase, usually a ZAI phrase, as complement. These prepositional phrases appear after the verb, and what is more special about them is that the preposition ZAI actually becomes part of the verb. When the sentence is stated in past tense, the auxiliary LE appears after ZAI instead of right after the verb, as in the example sentence in Figure 5-7. The way we solve this problem is by the “delayed pull” feature in GENESIS. The tense auxiliary is generated at the normal position, which is before the prepositional phrase but after a delayed pull, but it is put aside. Then the post-posed prepositional phrase picks it up. If there is no such prepositional phrase that picks up the tense auxiliary, then the delayed pull in the verb phrase will eventually get the auxiliary.

English: They stood under the tree
Chinese: 他们 站 在 了 树 下
They stand ZAI LE tree under

Figure 5-7 Example of preposition ZAI combined into the verb

Expletive. The subject “it” which exists only to satisfy syntax and doesn’t have semantic meaning, is called “expletive” in linguistics. The subjects of the examples in Figure 5-8 are examples of the expletive “it”. English sentences use quite a few expletives, especially when the real subject topic is long. But this is very rare in Chinese. In Chinese, the real subject topic always appears at the front, however long it is. So the Chinese translations of the first two sentences literally look like “To deliver it takes three days” and “To know this is important”.

a) English: It takes three days to deliver the letter.
Chinese: 送 信 要 三 天
deliver letter take three day
b) English: It’s important to know this
Chinese: 知道 这个 很 重要

know this very important

c) English: It is kind of you to say so
Chinese: 你 能 这么 说 太 好 了
You can so say too kind LE

Figure 5-8 Examples with expletives.

As the translations imply, the expletive actually causes very long distance reordering, so we definitely want to handle it in the linguistic system. However, even for the linguistic system it is a hard problem, because it is very difficult to decide which "it" is expletive only from syntax structure. Fortunately, after examining the corpus, we found that it is enough to focus on a small number of sentence patterns, more specifically, three kinds of sentences: "it take/cost … to …", "it is ADJ to …" where the ADJ is important, difficult, etc., and "it is ADJ of PRO to" where PRO is you, him, etc. We judge if a sentence is one of these three structures in the preprocessor of GENESIS. A series of checks are carried out to make sure the sentence has all the necessary components so that the subject can be determined as an expletive. Upon all the checks being positive, a flag is set to indicate that the sentence contains an expletive, as well as which constituent is the real subject topic. Then, in the generation, GENESIS checks the flag and yanks the real subject topic to the front.

Negation. We dealt with two situations about negation in which the Chinese translation is structurally different from the English. The first situation is when the negation happens in the object. When the object is modified by the negated quantifier "no", the negation is promoted to the verb. An example is given in (a) of Figure 5-9. The second situation is more complex. When the subject is negated, and is complex, such as a noun clause, the Chinese grammar doesn't allow a negated adjective to modify such a subject. Instead, the negation is expressed by a negated adverbial phrase in front of the verb, as exemplified in (b) in Figure 5-9.

a) English: I saw no trees.

Chinese: 我 没 看见 树

I not see tree

b) English: Nothing they said is good.

Chinese: 他们 说 的 没 一 件 是 好 的

They say NOM not one MEASURE is good NOM

Figure 5-9 Sentences with negation which are structurally different between English and Chinese.

Predicative Adjectives. Chinese has a special classification for adjectives: predicative and non-predicative. Besides as a modifier to the nouns, the predicative adjectives can function like a verb. They don't need a link verb in copular sentences, they can have tense auxiliaries and some of them can even take objects. Some examples are given in Figure 5-10. The predicative adjectives count for over half of the common Chinese adjectives, so we regard them as an important feature in the Chinese grammar, especially for beginning learners, and we have therefore tried hard to generate them correctly. We first borrow the list of predicative adjectives from the Chinese grammar for TINA and use the list to mark all the predicative adjectives in the lexicon of GENESIS. Then we eliminate the link verb "be" when the adjective complement is marked as predicative. Because they act as verbs, the tense auxiliaries keys and appropriate tense conversion flags are inserted to generate correct tense.

English: It is ready.

Chinese: 它 好 了

It ready LE

Figure 5-10 Predicative Adjectives in Chinese

Measure Words and Unit Nouns. Every noun in Chinese is associated with a measure word. We say "一本书"(a book) and "一辆车"(a car), but we cannot say "一辆书" or "一本车". The measure word can be determined by the property of the object, or by the type of container that the object is placed in. In the latter situation, the measure word is more like a unit noun, for example the word “cup” in “a cup of water”. While it might seem to be easier for a statistical system to choose a measure word by statistics, we want to do it in the linguistic system because of two reasons. The first reason is that the measure words are almost context independent. Once they are in the lexicon, they are correct except for very rare occasions. Secondly, the statistical system is less reliable in inserting a word than in deleting. And also, when the noun phrase contains a modifier like adjectives, or even long adjective clauses, the measure word will be far separated from the noun, which is very disadvantageous for the statistical system to produce the right measure word.

So we supply the measure word information in the lexicon by putting a :measure with a corresponding value for the nouns. It is not necessary to do this for all the nouns, because many nouns go with the general measure word “个”. In generation, when the measure word is required, we first check if there is a unit noun in the noun phrase. If not, the value of the :measure set by the noun is generated. If the :measure doesn’t exist at all, we back-off to the general measure word.

With all these careful manipulations of the generation rules and lexicon entries, many Zhonglish II sentences are actually perfect Chinese. The examples shown in Figure 5-6 to Figure 5-10 are the actual outputs after parsing the English and paraphrasing the semantic frame into Zhonglish II. Even if the Zhonglish II output is not perfect, we still have the statistical system which can potentially fix the translation. Because now the Zhonglish II sentences are much more like Chinese than the original Zhonglish, the statistical system can align the sentences with even fewer crossovers and generate even better models. As a result, it is much more promising that we can get perfect translations which can serve as reference translations for the students.

5.4 Automatic Assessment

The last piece of the language learning system is automatic assessment. The student definitely desires feedback from the system when he/she finished one translation. It is not kindly for the system to simply display the reference translation and say, "here is the correct answer. Please compare with your own answer." Translation is not a multiple choice. There may be many ways to translate a sentence correctly. If the system only shows the reference translation, the student will be confused about whether that is the only answer or maybe his/her answer is also correct. So the system must judge automatically for the student.

There are two aspects of the judgment. One is whether the student said a well-formed sentence, i.e., whether the translation conforms to the Chinese grammar. The other is whether the translation maintains the original meaning. The grammaticality can be tested by parsing. If the sentence can be parsed, it is at least probably syntactically well-formed. The test for meaning preservation is much more difficult. Since the same meaning can be conveyed in a variety of ways, string comparison obviously doesn't work. The meaning of a sentence is not only expressed by the words in the sentence, but also by the word order which forms the syntactical structure. So instead of comparing two strings, we compare the semantic frames produced by our parser. Because the semantic frame is a hierarchical structure, and contains richer information than the meaning comparison requires, we first flatten the semantic frame into a key-value sequence, and then the two key-value sequences are sent to an evaluator, which will compare and judge the similarity. Figure 5-11 illustrates the automatic assessment process.

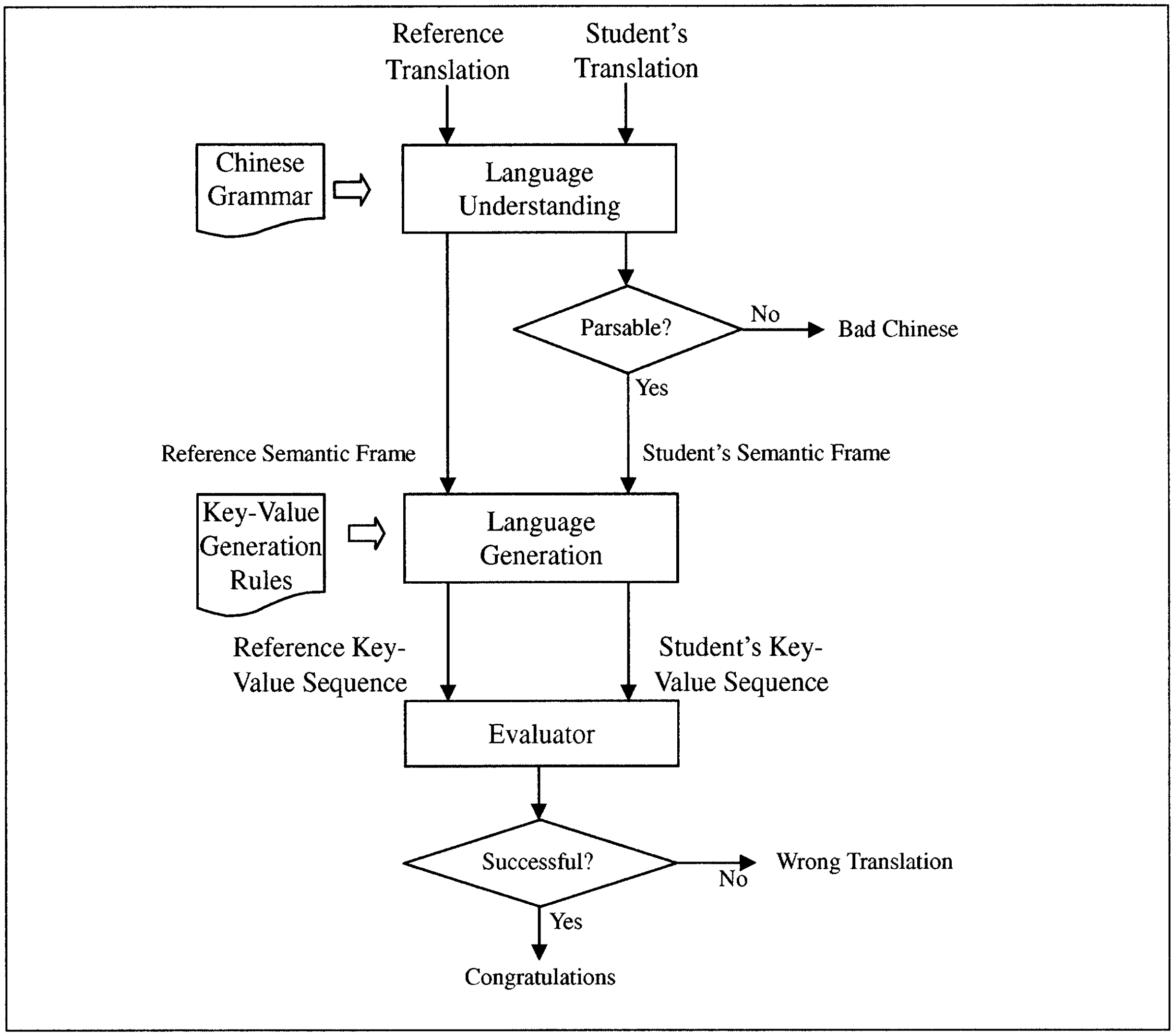


Figure 5-11 Process of automatic assessment

5.4.1 Generating Key-Value Sequence

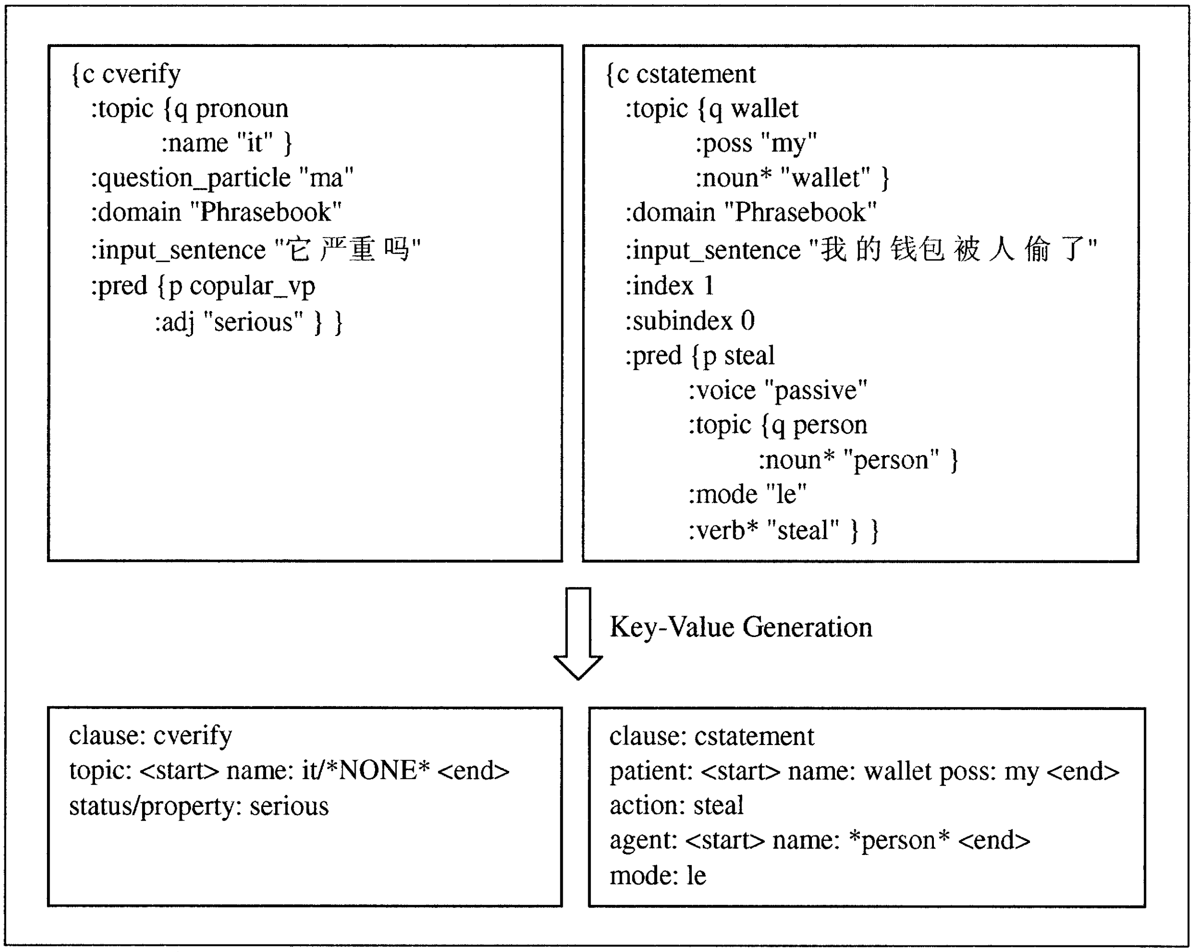
The process of generating the key-value sequence is essentially a conversion from semantic frame to string, so GENESIS is capable of doing this. We write the key-value generation rules to produce the key-value sequence. This set of rules is very different from the Zhonglish generation rules, or any other generation rules for natural languages. The output string is composed of a sequence of key-value pairs. The keys end with a colon. The values can be either a word (phrase), or another sequence of key-value pairs bracketed by special markers, in order to handle hierarchy.

The order of the key-value pairs doesn't matter at all. The goal is to represent the meaning of the semantic frame as simply as possible.

The key-value approach was also adopted by the previous two language learning systems. Because of the characteristics of the domains of those two systems, the key-value pairs can easily be identified. For example, in the flight domain, one can imagine several important keys, such as type of the sentence, flight number, departure time, etc. However, for the travel domain in this thesis, it is relatively hard to define such keys for particular events because the topics of the sentences are too broad. Instead, we adopt major semantic roles as the keys, for example agent, action, patient, time, location, color, etc. The values are extracted from the corresponding constituents in the semantic frame, and all the other information related to syntax only is discarded. Bracket markers are put around agent and patient, because they can have modifiers of the same kind, and the meaning will be confused without a bracket. Values for other roles are bracketed too when they are complex, for example adverbial clauses, adjective clauses, etc. We also allow some semantic roles to be optionally omitted by putting a special value "*NONE*", since Chinese has the property of omitting certain kinds of sentence components. By doing so, we compactly represent a semantic frame in a set of key-value pairs which are free of the original syntax of the sentence. A meaning expressed in active voice and passive voice will have the same set of key-value pairs ideally.

Now the difference in syntax structure is eliminated, but there is a problem for the semantics of the words. Many words, especially adjectives and adverbs, have synonyms. The meaning "good" can be expressed by a lot of words such as "great", "excellent", "wonderful" and so on. They should all look the same in the key-value representation in order to be evaluated correctly. We dealt with this problem by using the lexicon of GENESIS. The words are organized into their semantic classes, and all the members in one semantic class are mapped into one value in the lexicon. So, in the above example, adjective “great”, “excellent” and “wonderful” are all mapped into “good”. Adverbs, especially degree adverbs, are mapped into several categories too.

So finally we produce a set of simple key-value pairs with semantic classes from the original semantic frame. Some examples are shown in Figure 5-12.



```
{c cverify
  :topic {q pronoun
         :name "it" }
  :question_particle "ma"
  :domain "Phrasebook"
  :input_sentence "它 严重 吗"
  :pred {p copular_vp
         :adj "serious" } }
```

```
{c cstatement
  :topic {q wallet
         :poss "my"
         :noun* "wallet" }
  :domain "Phrasebook"
  :input_sentence "我 的 钱包 被 人 偷 了"
  :index 1
  :subindex 0
  :pred {p steal
         :voice "passive"
         :topic {q person
                :noun* "person" }
         :mode "le"
         :verb* "steal" } }
```

```
clause: cverify
topic: <start> name: it/*NONE* <end>
status/property: serious
```

```
clause: cstatement
patient: <start> name: wallet poss: my <end>
action: steal
agent: <start> name: *person* <end>
mode: le
```

Figure 5-12 Some examples of key-value pairs

5.4.2 Comparing Key-Value Sequences

We generate a key-value sequence for the reference translation, and another sequence for the student’s translation. The two sequences are sent into an evaluator to judge their similarity. The evaluator we use in this thesis is basically the one that was developed in our group to do key-value comparisons in other tasks, including the previous two language learning systems. The evaluator uses some heuristic algorithms, and will tell whether it is a perfect match, or how many insertions, deletions and mismatched keys there are between the two sequences. We only made a slight improvement to the algorithm. When the value for a key is a simple string rather than a bracketed value, we improved the evaluator by handling alternative choices. The motivation is that a word, typically a verb, can have different meanings when it appears in different scenarios. Instead of trying to judge which meaning it is in the particular sentence, the lexicon provides it with all the choices separated by slashes. When the evaluator compares two values, a match is considered when any of the choices matches the counterpart.

5.5 Evaluation

5.5.1 Reference Translation

We compared three systems for reference translation. The first one is the pure phrase-based statistical system. The second one is the combined system with the original intermediate language Zhonglish. The third one is the combined system with the improved intermediate language Zhonglish II. In all three systems, we trained the statistical model using the set selected by the average unigram score method in Section 5.2. The sentences that cannot be parsed by TINA on the English side were filtered out. This left us a set of 11,453 sentences. Similar as the evaluation done in Chapter 4, we looked at the GIZA alignment crossover on the training set. The results are shown in Table 5-2. Not surprisingly, the result from the third system has the least crossovers, about 78% fewer than the pure statistical system and about 18% fewer than the combined system with Zhonglish.

	Average Crossover	Average Normalized Crossover	Percentage of sentences with zero crossover
English-Chinese	10.19	1.42	17.9%
Zhonglish-Chinese	3.32	0.38	46.7%
ZhonglishII-Chinese	2.94	0.31	52.5%

Table 5-2 GIZA alignment crossovers on training set when trained with English-Chances pairs and ZhonglishII-Chinese pairs

Then we tested the three systems on the training set. Since we were testing on the training set, the results were expected to be the best performance of the systems. We calculated the conventional BLEU score on the translation outputs of the three systems. From Table 5-3, we see that both combined systems outperform the pure statistical system. But it is surprising that the English-Zhonglish-Chinese system is better than the English-ZhonglishII-Chinese system by 2 points. However, as stated in Chapter 4, this result might be affected by the fact that we only have one reference translation to compare with. And also, since BLEU score is the geometric mean of the n-gram precisions, it is still debatable whether it really correlates with the true translation quality for languages like Chinese, which have freer word order and more omissible words. Due to the uncertainty, we sampled 1/30 of the set, which is 382 sentences, and did a manual evaluation. The manual evaluation simply compares two translations produced by two systems, and marks which is better. We did this between the pure statistical system and the English-ZhonglishII-Chinese system, and between the English-Zhonglish-Chinese system and English-ZhonglishII-Chinese system. The results are shown in Table 5-4.

	BLEU
English-Chinese	85.96
English-Zhonglish-Chinese	90.35
English-ZhonglishII-Chinese	88.31

Table 5-3 BLEU scores on the training set when trained with English-Chinese pairs and ZhonglishII-Chinese pairs

	# of ZhnII-Chn is better	# of the other one is better
Comparing Eng-Chn and Eng-ZhnII-Chn*	51	13
Comparing Eng-Zhn-Chn and Eng-ZhnII-Chn*	26	12

Table 5-4 Human evaluation of the translation generated from three systems. * marks the results that are statistically significantly different.

The numbers in Table 5-4 clearly show that the human evaluator thinks the English-ZhonglishII-Chinese system got the most "better translations". By McNemar significance test, both results are significantly different. So we can say that, in fact, the English-ZhonglishII-Chinese system did the best among the three. We see how the linguistic system can help in improving the quality of the translations, and the more effort we put into the linguistic system, the better outputs we will get.

We also tested the three systems on another test set of 610 sentences, which is generated from the lesson templates. This is a totally independent test set, but since the templates are drawn from the training set, the sentence structures in this test set should be covered by all three systems. Because this time we don't have a reference translation, we only did a manual evaluation. The evaluation method is the same as above. Table 5-5 shows the results.

	# of ZhnII-Chn is better	# of the other one is better
Comparing Eng-Chn and Eng-ZhnII-Chn*	278	26
Comparing Eng-Zhn-Chn and Eng-ZhnII-Chn*	139	28

Table 5-5 Human evaluation of the independent test set generated by the lesson templates. * marks the results that are statistically significantly different.

The results show that the combined system with Zhonglish II did much better than the other two systems. Looking at the outputs from the three systems, we found that the combined system with Zhonglish II translated the sentences with similar structures most consistently. As shown in Table

5-6, when translating a couple of examples with only the color changed in the sentences, the English-ZhonglishII-Chinese system did fairly consistently even when the color word is out-of-vocabulary. In contrast, the pure statistical system produced translations in inconsistent structures. This demonstrates that we really are controlling the output structure by our linguistic rules.

English	Output from Eng-Chn	Output from Eng-Zhn-Chn	Output from Eng-ZhnII-Chn
I have a black car.	我有一个*黑色的车。 (I have a black car)	我有个*黑色的车。 (I have a black car)	我有一辆黑色的车。 (I have a black car)
I have a blue car.	我有一个*蓝色的车。 (I have a blue car)	我有一个*蓝色的车。 (I have a blue car)	我有一辆蓝色的车。 (I have a blue car)
I have a green car.	给我拿个*绿色的车。 (bring me a green car)	我有个*绿色的车。 (I have a green car)	我有一辆绿色车厢的。 (I have a green car)
I have an orange car.	给我拿个*orange车。 (bring me an orange car)	我有给我一杯*车。 (not grammatical)	我有个*orange车。 (I have an orange car)
I have a purple car.	给我拿个*purple车。 (bring me a purple car)	我有个*purple车。 (I have a purple car)	我有一辆purple的车。 (I have a purple car)
I have a red car.	我有红车。 (I have red car)	我有件*红色的车。 (I have a red car)	我有红色的一辆车。 (I have a red car)
I have a white car.	我有一个*白色的车。 (I have a white car)	我有个*白色的车。 (I have a white car)	我有一辆白色的车。 (I have a white car)
I have a yellow car.	给我拿个*yellow车。 (I have a yellow car)	我有个*yellow车。 (I have a yellow car)	我有一辆yellow的车。 (I have a yellow car)

Table 5-6 Comparison between the outputs from three systems. * marks the wrong measure word for noun "car".

There is another noticeable thing. When we examined the Zhonglish II sentences generated from the linguistic system, without going through the statistical system, we already saw many sentences that were in good shape. But then after the statistical step, they were no longer correct. We evaluated this situation manually on the 610-set, and obtained the result in Table 5-7.

	# of Zhonglish II is better	# of final output is better
Comparing Eng-ZhnII and Eng-ZhnII-Chn	84	81

Table 5-7 Manual comparison between ZhonglishII output and final output from statistical system

This almost random result is quite interesting. It reveals that, in a task which the input is constrained, and the training data is small, the linguistic system alone can do better than a combined system. The major reason is that the training data cannot cover all the possible instances generated from the templates, and thus will turn some good sentences into bad ones. However, this result is not disappointing. Instead, it is encouraging that we can have a simple linguistic system, or a linguistic system with much simpler statistical methods to finish the whole translation task. The advantage is that the whole translation component can run much faster, and occupy much less space. The linguistic system alone can translate the 11,453 sentences in less than two minutes to finish parsing and generation, while the statistical system will need over forty minutes to translate the same set. These features will make it more attractable when running stand-alone on a mobile device.

5.5.2 Automatic Assessment

Since the game system is not built and we do not have real subjects, we evaluate the automatic assessment in the following way. From the 610 sentences that the templates generate, we looked at the Zhonglish II translation, and manually selected 257 sentences that are perfectly translated. These good sentences serve as the references. Then, we pretended the three systems discussed in the previous subsection to be three students, and assess their performance. The translations from these pseudo students contain both perfectly good sentences and erroneous sentences. We then see how well the automatic assessment judges these translations according to the references.

		# of Correct Translations Judged by Automatic Assessment	# of Problematic Translations Judged by Automatic Assessment
Outputs from pure statistical system	# of Correct Translations Judged by Human	91	11
	# of Problematic Translations Judged by Human	2	153
Outputs from Eng-Zhn-Chn	# of Correct Translations Judged by Human	130	11
	# of Problematic Translations Judged by Human	5	111
Outputs from Eng-ZhnII-Chn	# of Correct Translations Judged by Human	145	17
	# of Problematic Translations Judged by Human	1	94

Table 5-8 Results of judgments from the system and the human

We can see that the human judge and the system agree fairly well. The system made correct judgments for over 90% of the time. We also examined the translations from the third pseudo student for which the human judge and the system disagreed more closely. We found that the translation which the system considered as good, but the human judge considered as problematic, has a very similar surface form as the reference translation. It is syntactically correct, but the meaning has subtle differences with the reference, which are really hard to judge. Among the 16 translations that the system misjudged as problematic, 11 had parse failures because of the coverage of grammar, and the remaining 6 are due to the fact that our key-value representation cannot represent the two ways of expressing in the same form. For the first situation, it can potentially be solved as we continue to improve Chinese parsing. For the third case, we can devote more effort to improving the key-value representation, and also improve the comparison algorithm to make it focus more on the semantics but not the structure. We also see situations in the other two test sets where the automatic assessment and the human judge disagree because of nativeness. This is very hard to judge, since it relates to factors that are not syntactical or semantical. But we can try methods like language models to help make the decision.

Chapter 6. Conclusion and Future Work

In this thesis, I talked about a translation game system for learning Mandarin Chinese in the travel domain. The system asks the user to translate randomly generated English sentences into Chinese, and provides automatic feedback on the correctness of the translations. The system is challenging because English and Chinese are two languages with large differences, and also because the domain we are dealing with is relatively open. The framework of the game system is leveraged from two previous language learning systems developed in our group, but we made crucial improvements to two main components: the Chinese parsing component, and the English-Chinese machine translation component.

The problem with parsing Chinese is that both the speed and the parse coverage was low. This is because our parser TINA was designed for top-down languages such as English, but Chinese contains many bottom-up characteristics. We solved this problem by adding the restructure feature, a bottom-up feature, into TINA. The restructuring consists of two operations: look-left-and-absorb and look-up-and-rename. Both restructure operations are aimed at reducing redundant parsing. Part of the parsing decisions are delayed until the initial parse tree is built, so that the common prefix of the phrases, and the phrases with similar structures can be parsed efficiently in one hypothesis. We also added another kind of feature, "parasites" and "enemies", to deal with long distance non-context free constraints. The parasites and enemies make it easy to tackle special cases in the grammar, and help the grammar to be more precise. Great improvements were gained with these new features. The parse coverage rose from around 65% to over 80%, with only about one tenth of original time usage.

For the English-Chinese machine translation, since the domain is relatively open, pure linguistic method would need vast manual effort. Pure statistical method would face problems, too, because there are large differences between English and Chinese, and the size of our corpus is very small compared to the typical size of training corpora in the statistical machine translation field. So we attempted to combine the linguistic and statistical systems. We first use the linguistic system to translate the English sentences into a man-made intermediate language "Zhonglish", in which most of the words are still in English, but the word order obeys Chinese grammar as much as feasible. Then, we use the standard phrase-based statistical machine translation system to translate Zhonglish into Chinese. In doing so, we put together the advantages of both linguistic and statistical systems, and avoided their weakness. Most of the sentence construction is done by the linguistic system, which is more reasonable and controllable than the statistical system. On the other hand, lexical mapping is mostly dealt with by the statistical system, in which the statistics can help a lot with the word-sense disambiguation. In the combined translation systems, we also developed techniques to deal with random constructions, phrases that can be fully pre-translated by the linguistic systems, and ambiguous parses. The experiment showed that our combined translation system outperforms the baseline statistical system by over 2 BLEU point, and has 45% fewer GIZA alignment crossovers.

Finally, we adapted the above two main components into the translation game system. We ranked the sentences in the corpus by word unigram scores, and selected about 12,000 sentence pairs which contain the 1,000 most frequent Chinese words. Templates were built from these sentences. Then, in the step of reference translation generation, we further improved the intermediate language Zhonglish into Zhonglish II, in order to provide high-quality translation. Zhonglish II contains actual Chinese characters, and conforms to Chinese grammar more closely. The lexicon for Zhonglish II was obtained in a semi-automatic way. From the experiments, we can see that this combined version of translation system did the best compared to the pure statistical system and the combined system with the original Zhonglish, especially in the task where the input sentences are generated from the templates. Lastly, in the meaning comparison step, we generated key-value pair sequences from the semantic frames of the reference translation and the student's translation. The key-value pairs extract the important semantic roles from the semantic frames and ignore the differences in the syntax level. The experimental results showed that this method of automatic assessment agrees with the human judge for over 90% of the test sentences.

There are quite a few of things that be done for future work. These can be summarized into three aspects. The first one is the real game interface. We didn't build the actual Web interface in this thesis due to time limitations. The technologies are ready, so it should not be hard to build the interface. With the actual system in place, we can have real data from subjects and do more analysis. The second aspect is further improvement of the Chinese grammar and Zhonglish generation rules. The Chinese grammar has been improved a lot, but we still see much possibility for further improvement. This is also true for Zhonglish, mainly the Zhonglish II generation rules. The last aspect is to try to lessen the time and space requirements for the whole system, so that it can be migrated into mobile devices.

References

Yasuhiro Akiba, Eiichiro Sumita, Hiromi Nakaiwa, Seiichi Yamamoto, Hiroshi G. Okuno. 2003. *Experimental Comparison of MT Evaluation Methods: RED vs. BLEU.* In Proceedings of MT Summit IX, New Orleans, USA, 2003.

Lauren Baptist and Stephanie Seneff. 2000. *Genesis-II: A Versatile System for Language Generation in Conversational System Applications.* In Proceedings of ICSLP, Beijing, China, 2000.

Marine Carpuat and Dekai Wu. 2007. *Improving Statistical Machine Translation using Word Sense Disambiguation.* In Proceedings of EMNLP, Prague, Czech Republic, 2007.

Chi-Hyu Chao, Stephanie Seneff, and Chao Wang. *An Interactive Interpretation Game for Learning Chines.* Proc. of the Speech and Language Technology in Education (SLaTE) Workshop, Farmington, Pennsylvania, 2007.

Boxing Chen, Jun Sun, Hongfei Jiang, Min Zhang and Ai Ti Aw. 2007. *I^2R Chinese-English Translation System for IWSLT 2007.* In Proceedings of IWSLT, Trento, Italy, 2007.

Brooke A. Cowan. 2004. *PLUTO: A Preprocessor for Multilingual Spoken Language Generation.* Master's Thesis, MIT, Cambridge, Massachusetts, 2004.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, Evan Herbst. 2007. *Moses: Open Source Toolkit for Statistical Machine Translation.* Annual Meeting of the Association for Computational Linguistics (ACL), demonstration session, Prague, Czech Republic, June 2007

Daniel Marcu, Wei Wang, Abdessamad Echihabi and Kevin Knight. 2006. *SPMT: Statistical Machine Translation with Syntactified Target Language Phrases.* In Proceedings of EMNLP, Sydney, Australia, 2006.

Stephanie Seneff. 1992. *TINA: A Natural Language System for Spoken Language Applications.* Computational Linguistics, Vol. 18, No. 1, 1992.

Stephanie Seneff and Joseph Polifroni. 2000. *Dialogue Management in the MERCURY Flight Reservation System.* In Proceedings of ANLP-NAACL, Satallite Workshop, Seattle, WA, 2000.

Wade Shen, Brian Delaney and Tim Anderson. 2006. *The MIT-LL/AFRL IWSLT-2006 MT System.* In Proceedings of IWSLT, Kyoto, Japan, 2006.

Chao Wang, Michael Collins, and Philipp Koehn. 2007. *Chinese Syntactic Reordering for Statistical Machine Translation.* In Proceedings of EMNLP, Prague, Czech Republic, 2007

Chao Wang and Stephanie Seneff. 2007. *A Spoken Translation Game for Second Language Learning.* In Proceedings of AIED, Marina del Rey, California, 2007

Chao Wang and Stephanie Seneff. 2006. *High-quality Speech Translation in the Flight Domain.* In Proceedings of Interspeech, Pittsburgh, Pennsylvania, 2006.

Kenji Yamada and Kevin Knight. 2001. *A syntax based statistical translation model.* In Proceedings of ACL 2001.

Dongdong Zhang, Mu Li, Chi-Ho Li and Ming Zhou. 2007. *Phrase Reordering Model Integrating Syntactic Knowledge for SMT.* In Proceedings of EMNLP, Prague, Czech Republic, 2007.

Ying Zhang, Stephan Vogel, Alex Waibel. 2004. *Interpreting Bleu/NIST scores: How much improvement do we need to have a better system?* In Proceedings of LREC, Lisbon, Portugal, 2004.

Victor Zue, Stephanie Seneff, James Glass, Joseph Polifroni, Christine Pao, Timothy J. Hazen and Lee Hetherington. 2000. *JUPITER: A Telephone-Based conversational Interface for Weather Information.* IEEE Transactions on Speech and Audio Processing, 8(1): 85-96.

Appendix A. Syntax for Frame Transformation Rules

The function of frame transformation rules is to rewrite the input frame into the output frame. The frame transformation rules are written in the GALAXY frame format and take effect sequentially. Each rule has three necessary keys: *:in*, *:replace*, and *:with*. The *:in* key specifies the condition to trigger the rule. The *:replace* key specifies what to replace, and the *:with* key specifies the desired outcome. There is another optional key *:copy*. It has the function of copying the values of a list of keys from the input semantic frame for later use. Following are two examples. The left example simplifies a copular_vp predicate frame. The right example copies the negation from the predicate frame into the upper clause frame.

```
{c transformation_rules
        :in {p copular_vp
                :pred {p prep_phrase
                   :trace “where/which” } } }
        :replace “*SELF*”
        :with {p copular_vp
                :trace “where } }
```

```
{c transformation_rules
        :in {c *ANY*
                :pred {p *ANY*
                   :negate “*ANY*”
                   :*submatch* 1 } }
        :copy “:negate”
        :replace “*SELF*”
        :with {c *SELF*
                :negate “*COPY*” } }
```

Figure A-1 Two examples of frame transformation rules

In the following description, the semantic frame to process is called the input frame. The semantic frame after applying the transformation rules is called the output frame.

A.1 *:in* key

The value for *:in* key is a frame (the frame will be called the in_frame in the following). The in_frame describes the condition to trigger the transformation. There are some special values and keys that can be used in the in_frame.

- **Value **ANY**.** A frame with the name **ANY** will match any frame with the same type (clause, predicate, or topic). A key with value "**ANY**" will match any value as long as the key exists.

- **Value **NONE**.** A key with this value means that the input frame does not contain this key. Match will fail if the key is found in the input frame.

- **Negation.** A frame name starting with *NOT_* stands for negation. For example, {q NOT_pronoun} means to match any topic frame except the pronoun frame.

- **Key *:*submatch**.** If *:*submatch** appears and the value is set to integer *1*, the matching procedure will be a submatching for the current frame. Only the keys specified will be checked, but there can be additional keys in the source semantic frame. Without the *:*submatch** key, the input frame cannot have additional keys. The top level of the in_frame contains a *:*submatch** key with value 1 implicitly. The *:*submatch** for child frames needs to be specified explicitly.

- **Key *:*focus**.** This key specifies which frame is the focus frame to replace. By default, the focus frame is the top level frame. But in the situation that the top level frame is the parental condition, for example, to replace a predicate frame A under predicate frame B, it is necessary to set the *:*focus** key with integer value *1* within the child frame.

A.2 *:replace* Key

The value for the *:replace* key is a string. There are two choices. The first one is a key name which appeared in the in_frame. The semantics is to replace the value of that key. In this case, the value for the :with key can only be a string. The other one is a special string **SELF**, which means to replace the entire focus frame.

A.3 *:copy* Key

The value for *:copy* key can be either a string, or a list of strings. Each string should be the key name that appeared in the in_frame.

A.4 *:with* Key

The value for *:with* key can be either a string or a frame. If it is a string and the value for *:replace* is *"*SELF*"*, the action is to rename the frame. If it is a string and the value for *:replace* is a key name, the action is to set the value of that key to the string.

If the value for :with is a frame (the frame will be called with_frame in the following), the value for *:replace* key must be *"*SELF*"*. There are two choices of a with_frame. If the name of the with_frame is **SELF**, the frame operations will be done on the input frame. If the with_frame has a concrete name, it will create a brand new frame in the output frame and discard the input frame. The following example demonstrates the difference.

Input Frame	{c clause :auxil “can” :topic “I” }	
Transformation Rule	{c transformation_rules :in {c clause } :replace “*SELF*” :with {c *SELF* :auxil “should” } }	{c transformation_rules :in {c clause } :replace “*SELF*” :with {c clause :auxil “should” } }
Output Frame	{ c clause :auxil “should” :topic “I” }	{c clause :auxil “should” }

Table A-1 Comparison between two types of with_frame

There are several special actions that can be used in the with_frame.

Deletion. To delete a key, use value “*NONE*”. For example _:auxil “*NONE*”_ means to delete the key :auxil. To delete a predicate, add “_DEL__” to the front of the predicate name. For example, _{p DEL_adjective}_ means to delete the predicate named adjective. Deletion should only happen when the name of the with_frame is _*SELF*_.

Add Rest. The special key :*rest* with integer value 1 can put all the keys in the focus frame that are not mentioned in the in_frame into the output frame. This is useful when both the renaming of the frame and the modification of the content are desired.

Paste the Copied Values. To paste the values that have been copied by _:copy_ key, use the string value _“*COPY*”_. If the _:copy_ key contains a list of key names, use _“*COPY-1*”_, _“COPY-2*”_, … in sequence to access those values.

A.5 More Examples

The following transformation rule rewrites a frame with subject “we” and Chinese ending “ba” into a “let’s” request.

Input frame	{c cstatement
	:topic {q pronoun
	:name “we” }
	:pred {p go}
	:ending “ba” }

Transformation Rule	{c transformation_rule :in {c cstatement :ending "ba" :topic {q pronoun :name "we"} } :replace "*SELF*" :with {c crequest :ending "*NONE*" :lets "let's" :topic "*NONE*" :*rest* 1} }
Output frame	{c crequest :lets “let’s” :pred {p go} }

Table A-2 Example transformation rule 1

The following transformation rules create a confirm clause.

Input frame	{c cstatement
	:topic {q car :quantifier “def” }
	:auxil “will”
	:pred {p move}
	:confirm_clause “confirm_clause” }

Transformation Rules	{c transformation_rule :**remark "create a clause" :in {c cstatement :confirm_clause "*ANY*" } :replace "*SELF*" :with {c *SELF* :confirm_clause {c confirm_clause} } } {c transformation_rule :**remark “copy the auxil” :in {c cstatement :auxil "*ANY*" :confirm_clause {c confirm_clause :*focus* 1} } :copy ":auxil" :replace "*SELF*" :with {c *SELF* :auxil "*COPY*"} }

	{c transformation_rule :**remark “no auxil, then create a xdo” :in {c cstatement :confirm_clause {c confirm_clause :auxil "*NONE*" :*focus* 1} } :replace "*SELF*" :with {c *SELF* :auxil "xdo"} } {c transformation_rule :**remark “add negate if the main clause is not negated” :in {c cstatement :negate "*NONE*" :confirm_clause {c confirm_clause :*focus* 1} } :replace "*SELF*" :with {c *SELF* :negate "not"} } {c transformation_rule :**remark “create topic it if the topic of main clause is not pronoun” :in {c cstatement :topic {q NOT_pronoun} :confirm_clause {c confirm_clause :*focus* 1} } :replace "*SELF*" :with {c *SELF* :topic {q pronoun :name "it"} } } {c transformation_rule :**remark “copy the pronoun topic from main clause” :in {c cstatement :topic {q pronoun} :confirm_clause {c confirm_clause :*focus* 1} } :copy ":topic" :replace "*SELF*" :with {c *SELF* :topic "*COPY*"} }

Output frame	{c cstatement
	:topic {q car :quantifier “def” }
	:auxil “will”
	:pred {p move}
	:confirm_clause {c confirm_clause
	:topic {q pronoun :name “it”}
	:auxil “will”
	:negate “not” } }

Table A-3 Example transformation rule 2