A Unification-based Approach to Lexicography for Machine Translation System

Shu-Chuan Chen*, Mei-Hui Wang*, and Keh-Yih Su**

***BTC R&D Center**
2F, 28 R&D Road II
Science-based Industrial Park
Hsinchu, Taiwan, R.O.C.

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

ABSTRACT

In an operational machine translation system, a variety of texts will be encountered even if its domain of dexterity is restricted to a specific field. This diversity of texts poses problem on handling different usages or translations of identical lexical items. This paper presents a unification-based method for lexicography that can greatly lessen this problem. In the paper, we give a detailed discussion and example of the unification technique. We also show that by unifying lexical information in different dictionaries, the time spent in dictionary construction is saved; dictionary storage space is minimized; the integrity of distinct dictionaries is preserved; the option regarding which dictionaries to be unified is kept open; and all of the lexical information needed to construct a complete dictionary based on the vocabulary for a specific customer project is available. In view of the fact that categorial ambiguity might occur as a result of unification, score function is added as a solution. With these advantages, we regard the unification approach to lexicography as viable in enhancing the translation performance of a practical machine translation system.

1. Introduction

In an operational machine translation (MT) system, even if its domain of usefulness is restricted to a specific field[1], a rich variety of texts will still be encountered. For instance, if the domain is limited to articles on computer science, texts in the areas of user manuals, programming languages, hardware, etc. are all possible inputs. These texts may differ in the use of individual words, the size of glossaries, the patterns of syntactic constructions, and so on.

For an operational system like ArchTran, which is a commercialized English-Chinese machine translation system developed at BTC R&D Center, the main concern in the face of diversity of texts is the ability to deal with different usages or translations of identical terms[2].

The problem concerning different usages or translations of identical terms is two-fold. On the one hand, different usages or translations may result from **ambiguity in word sense.** On the other hand, the differences may be due to the **requirements of customers.**

The problem concerning word sense ambiguity is that a good number of words have more than one possible meaning, and different meanings may call for different translations. For example, the word *current* may be in the sense of "water flow" in one text, and "electricity flow" in another. The former use of the word will be translated into Chinese as "水流", and the latter as "電流" accordingly.

To disambiguate the semantics of a polysemous word found in a text in order to render the correct translation, the following knowledge sources should be incorporated into the MT system: morphological information (a word used as a countable or uncountable noun may mean differently); syntactic information (different internal arguments may give rise to different meanings of a verb); semantic information (selectional restrictions); and pragmatic or contextual information (using the technique of "script"). Nevertheless, for a second generation MT system like ArchTran, not all the information needed for disambiguation is available or complete [Boit87]. Therefore, other means of disambiguation have to be incorporated as well.

In the ATLAS-G system [Fuji89], finding the correct translation, i.e. meaning, is in part done interactively by selecting and remembering the most appropriate translation for a given word in a given text. The problem with this approach is that once chosen, a translation will be assumed for the rest of the text. If the selected translation is suitable for just a few occurrences of the word, the translation of the other occurrences will be in error.

As for the problem of satisfying customer's requirements, a customer may wish a specific translation for a term, and the MT system must be able to do that. For example, one customer may prefer the term *operating system* to be translated as "作業系統", while another as "操作系統".

An obvious solution to the problem of satisfying a customer's request of a specific translation is to change the translation listed in the system dictionary into the one preferred by

the customer each time a text is translated. This, however, is problematic, since the translation has to be changed from time to time to be in compliance with a particular text. Besides, if more than one text is being translated at the same time, the change may be suitable for the word in one text but not the others.

As an alternative solution, one may propose to construct a separate and self-contained dictionary for each text. However, chances are the glossaries of different texts may differ only in a relatively small number of words, thus building separate dictionaries is not feasible. Because by doing so, the time spent on lexicography and the storage space taken up by the dictionaries with a huge amount of shared words and lexical information are wasteful.

Another possible solution is to create a run-time dictionary that stores only those words whose meanings or translations are specified by the user interactively, and the life span of the dictionary lasts just for the text currently under translation. This method suffers the same drawback as the ATLAS-G system. Furthermore, because a run-time dictionary is not accessible to other texts being translated at the same time and also because it is not accessible to similar texts to be translated at a later time, the power of a time-sharing computer is not fully utilized.

Discussions on disambiguating word senses abound in the MT literature [Alle87, Hirs87, Hutc86, Nire87]. These discussions focus mainly on the use of the various knowledge sources mentioned before. A second generation MT system, as pointed out, is limited in its access to these knowledge sources. On the other hand, little discussion can be found on the issue of producing translations preferred by customers. The solutions examined above concerning "customer tailored" translations are unsatisfactory. In this paper, a unification approach to dictionary information combination is proposed as a new and viable way to deal with different usages and translations of identical words that occur as a result of diversity in texts. The unification technique has been implemented in the ArchTran system and proved to be of fruitful result.

2. Principles in Constructing ArchTran Dictionaries

The way in which the ArchTran dictionaries are constructed and the way in which the dictionary information is unified during parsing are the key to solving the problem of different usages and translations of identical words. Hence, before going into the details of the use of unification, a brief introduction of the principles behind lexicography in the ArchTran system is in order.

Below are some of the major principles governing dictionary construction in ArchTran:

Principle 1 : Use all possible information, whether morphological, syntactic, or semantic, to disambiguate word senses of a lexical item. The corresponding translations of these senses are recorded in the dictionary.

Principle 2 : Create separate dictionaries to store words used in different domains and for different customers. No duplicate information is allowed in these dictionaries.

Under this principle, ArchTran developed three types of dictionaries. One is constructed to store words that can be found in all sorts of texts, called **general dictionary**. The second dictionary is called **technical dictionary**, which encompasses the words used in a particular field, such as machinery, computer science, etc. The third is a **customer dictionary** for storing technical terminology that differs from or lacks in both the general and the technical dictionaries. That is, the terminology in the customer dictionary is specific to the texts of a particular customer. It should be noted that for a given technical domain, there may be more than one customer dictionary, because customer dictionary can be further branched into several sub-dictionaries to store terms for different customers or for different projects.

Consider the following example that illustrates the functions of the three types of dictionaries. The word *computer* will be listed in the general dictionary for its established usage in nearly every walk of life. The word *firmware*, used solely in the domain of computer science, will be listed in the computer technical dictionary. And the word *Macrokey*, a term denoting a software package developed by BTC that enables users to define their keyboard functions, is listed in the customer dictionary for a particular project.

It should be noted that the very criteria that determine in which dictionary a word should be stored also regulate other information of a word, such as categories, word senses, internal arguments of verbs, and so on. For instance, suppose that a lexical item has three distinct word senses: A, B, and C. If A can be found in the texts of various fields, it is stored in the general dictionary. If B is used in the field of computer science solely, it is stored in the computer technical dictionary. And if C is used exclusively in the texts of a specific company or project, it is stored in a customer dictionary accordingly.

These three types of dictionaries and their sub-dictionaries are organized in a hierarchy by generality. In the case that a term is listed in more than one dictionary, its use is supposed to be most specific in the texts of a specific customer project, less in a technical domain, and least in general use. The hierarchical structure of the general, technical, and customer dictionaries in the ArchTran system are illustrated in Figure 1.

Building three different types of dictionaries serves several important purposes. The first and the main purpose is to **render the most suitable translation** for a polysemous word or to meet customer's requirement. If the MT system can not successfully disambiguate the senses of a word using all the knowledge sources noted in Principle 1 above, restricting the domain of translation will be of help. The accuracy in disambiguating the semantics of a word can be enhanced, since in a specific domain, the number of possible meanings of a polysemous word is, in most cases, limited. And only this limited number of meanings needs to be differentiated and recorded in the dictionaries. Since the meaning or translation listed in the customer dictionary is the most likely one to be used in the texts of a specific domain than

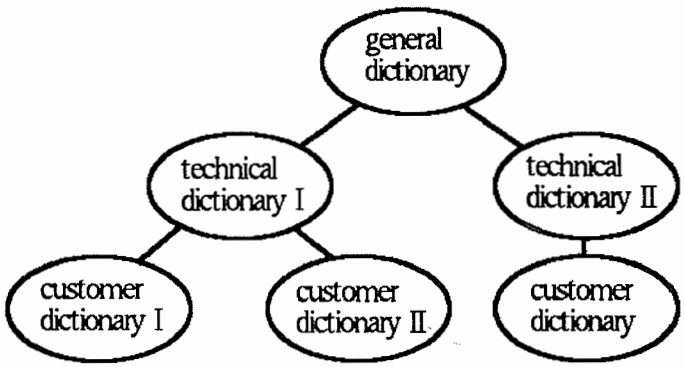

Figure 1 : The hierarchical structure of the general, technical, and customer dictionaries in terms of generality

that in the technical dictionary, therefore, during translation it has the priority of being applied before that in the technical dictionary. The same holds for the entry in the technical dictionary and the general dictionary. Thus, the most suitable meaning, or translation, can be correctly produced by this priority ordering. This point will be further exemplified in Section 3.2

The second purpose is to **save dictionary construction time**. For an MT system to translate texts of different fields, it is important to build separate technical and customer dictionaries to store the terminology. As each dictionary is defined as to the kind of lexical items and lexical information it should store, no information will be duplicated in these dictionaries. Thus, eliminating duplicate storing of the same data will make dictionary construction time-efficient.

The third purpose of building three types of lexicons is to **save dictionary storage space**. The storage space taken up by dictionaries is significantly cut down, since each dictionary stores no more words or lexical information than it is purported to.

The last purpose is to **maintain integrity of dictionary**. The integrity of the general dictionary and technical dictionary can be maintained, since no changes will be made directly on the lexical items in these dictionaries every time a particular translation is preferred by a customer.

As there are three types of dictionaries in ArchTran, and each stores no more lexical items or lexical information than is specified, the need of unifying these dictionaries is obvious during translation, because only by unifying these dictionaries can the most suitable translation be obtained and a complete set of glossaries be available.

In the next section, the technique of unification will be discussed at length.

3. Unification Operation in ArchTran

3.1 Unification in Lexicography

Unification is an operation employed in quite a number of linguistic and computational theories. Basically, unification is similar to the notion of set union when the elements to be unified are atomic elements. Unification departs from set union when unifying complex-feature-based information elements. Unification is said to "fail" when the values of the same features to be unified clash; the operation succeeds when the values of the same features match. If unification succeeds, the "merge" operation may be subsequently performed [Shie86, Huan88].

The example below shows how unification is used in ArchTran for lexicography. Provided that there are two dictionaries in ArchTran that have an identical entry LEX. We will call the LEX in these two dictionaries LEX1 and LEX2, respectively. Let us assume that LEX1 and LEX2 differ in both category and meaning (they may differ in other aspects and the same principle applies), and the lexical information of LEX1 and LEX2 can be notated by feature structures as shown in Figure 2:

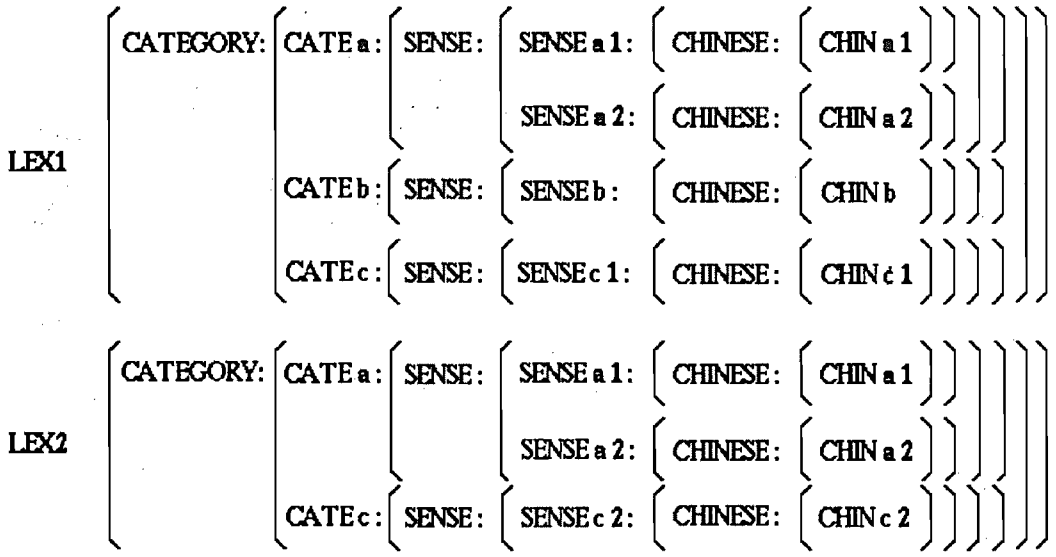

Figure 2: Differences between LEX1 and LEX2

In Figure 2, CATEGORY is a feature. CATEa, CATEb, and CATEc are values of CATEGORY and are features themselves. SENSE is the value of CATEa, CATEb, and CATEc and is a feature itself. SENSEa1, SENSEa2, SENSEb, SENSEc1, SENSEc2 are values of SENSE and are features as well. CHINESE is the value of SENSEa1, SENSEa2, SENSEb, SENSEc1, SENSEc2 and is a feature itself. CHINa1, CHINa2, CHINb, CHINc1, and CHINc2 are

The result of unifying LEX1 and LEX2 is shown in Figure 3:

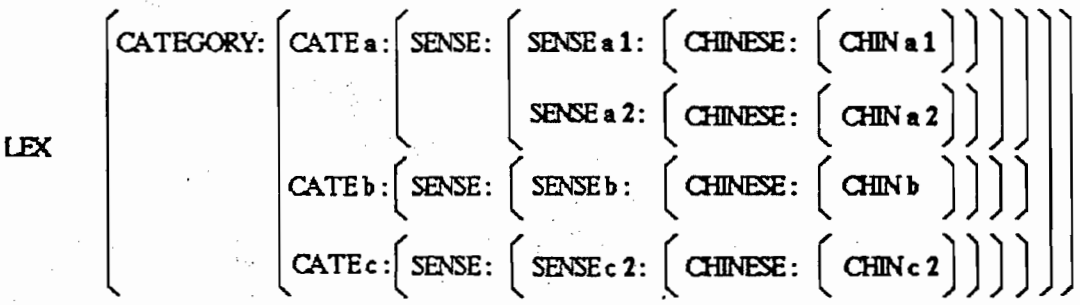

Figure 3 : Result of unifying LEX1 and LEX2

From Figure 2 and Figure 3, we can see that for the feature CATEa, the values in LEX1 and LEX2 match with each other and can be subsequently merged. As for CATEb, since there is no counterpart in LEX2, it will be included. For CATEc, the values of SENSEc1 and SENSEc2 are in conflict and, as a result, unification fails. In ArchTran, an important operation when unification fails is **overwriting** [Shie86], by which we mean that the lexical information in one dictionary will replace that of the other. In this case, SENSEc2 in LEX2 overwrites SENSEc1 in LEX1.

A question that arises here is that which dictionary has the right to overwrite. As noted above, ArchTran has three types of dictionaries: customer dictionary, technical dictionary, and general dictionary, and they are organized in a hierarchy by generality. Therefore, for a given lexical item, as its use is most specific in the texts of a specific customer project, less in a technical domain, and least in general use, its data in the customer dictionary overwrite those in the technical dictionary, which in turn overwrite those in the general dictionary.

In the following section, we will give a concrete example to illustrate the use and effect of unification in the ArchTran system in combining the information of all the entries of a term found in different dictionaries.

3.2 An Example

Consider the word *stream*. It can be used as noun and verb, and both categories are stored in the general dictionary. One of the meanings of the noun is "brook", and its corresponding Chinese translation is given as "溪流". One of the meanings of the verb when used as a transitive verb is "cause to flow", and its corresponding Chinese translation is given as "使流出". Provided that other senses of *stream* are not distinguished in the system, these are the only two senses of the word listed in the general dictionary.

In the field of computer science, the same word when used as a noun means "stream of data", and it is translated into "資料流". When it is used as a transitive verb, it means "execute sequentially" and is translated into "依序執行". These two senses are recorded in the technical dictionary for computer domain.

Now the lexical information of the word *stream* recorded in the technical and the general dictionaries is as shown in Figure 4:

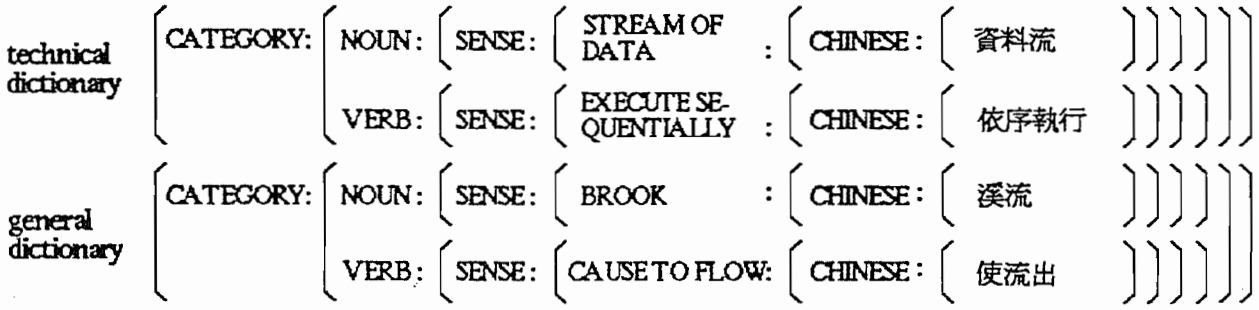

Figure 4 : Lexical information of the word "stream" in the technical and the general dictionaries

Suppose that no translations are specified regarding the translations of both the verb and the noun, after unifying the two dictionaries, the resultant lexical information of *stream* used in translating a text in the domain of computer is shown in Figure 5:

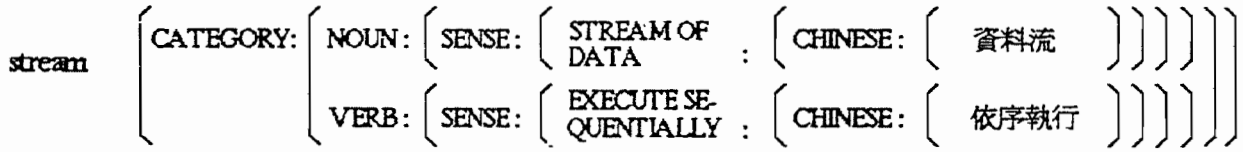

Figure 5 : Lexical information of the word "stream" after unifying the technical and the general dictionaries

As can be seen in Figure 5, the translation of *stream* when used as a verb will be "依序執行" and when used as a noun will be "資料流", rendering the most suitable translations for the word used in a text in computer science.

Suppose that in translating computer user manuals for a customer, the verb *stream*, used to mean the same as that in the technical dictionary, is preferred to be translated as "執行". This customized translation will be stored in the customer dictionary. Now the lexical information of the word *stream* listed in the customer dictionary is as shown in Figure 6:

Figure 6 : Lexical information of the word "stream" in the customer dictionary

After unifying the three dictionaries, the resultant lexical information of *stream* used in translating the text is shown in Figure 7:

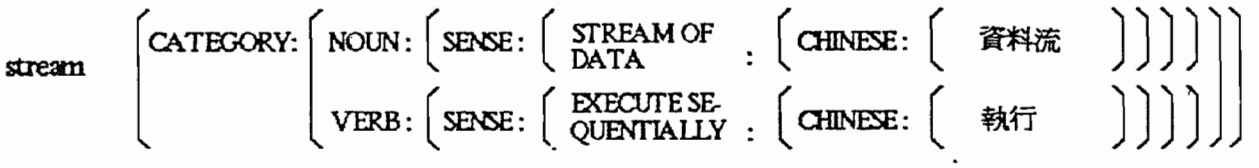

Figure 7 : Lexical information of the word "stream" after unifying the customer, technical, and general dictionaries

As can be seen in Figure 7, the translation of *stream* when used as a verb will be "執行", meeting the requirement of the customer.

In the following section, we will discuss the merits of employing unification in dictionary information combination.

4. Merits of Using Unification in Lexicography

4.1 Merits of Unification

Besides satisfactorily handling the problem of different usages and translations of identical terms found in various texts, there are at least three other advantages of using unification in dictionary construction:

[1] **Providing customized vocabulary without resorting to run-time dictionary** : The use of unification has the same effect as a run-time dictionary in providing customized vocabulary, but it does not have its drawback noted in Section 1; that is, a run-time dictionary is accessible only to one single text running at a given time.

[2] **Providing options in unifying various dictionaries** : We can specify which technical dictionary and which customer dictionary to be unified with the general dictionary for each text to be translated. Thus, the system dictionaries can handle texts of different fields and from different customers.

[3] **Complete dictionary available based on the vocabulary for a customer project** : A seeming disadvantage of employing unification is that the customer dictionary is not self-contained, since it consists only of those words or data that are distinct from those in the other two types of dictionaries. This problem can be easily solved by unifying all the dictionaries into one. And a complete dictionary is available if needed.

These advantages support the use of unification in lexicography. Nevertheless, there is a problem with the effect of unification. This consequence and its remedy will be examined in the next section.

4.2 Resolution of Categorial Ambiguity Resulting from Unification

As discussed in Section 2, the question as to in which dictionary a specific word should be stored is determined by the domain where it appears. In other words, only in a particular field, a particular attribute of a word is likely to appear (of course, it is not absolutely certain as to where a particular use of a word will definitely appear or not appear). Thus unifying dictionaries sometimes brings about more categorial ambiguities than is desired. For example, if the use of the word *default* as a verb is dominant or solely in a particular text, the category is consequently stored in the corresponding customer dictionary. Suppose that the more commonly used category noun is stored in the general dictionary, by unifying the two dictionaries, categorial ambiguity will probably result when translating a text.

ArchTran has devised a method to handle this problem by examining a portion of the text before translation, and then assigning a score to the category of a lexical item in the customer dictionary (or the one in the technical dictionary, if the word has no entry in the customer dictionary) relative to the category in the technical or the general dictionary. The one with a higher functional frequency score will suppress those with a lower score. To continue with the example in Section 3.2, suppose that by examining part of the text to be translated we find that for the word *stream*, verb is the dominant category, then the verb in the customer dictionary will be given a higher score than the noun in the technical dictionary. Thus, during translation the verb will be chosen for *stream* if both categories are found in the output structures, or ambiguous parse trees, for a sentence. If the fact has been shown to be the contrary, then the verb in the customer dictionary will be given a lower score than the noun in the technical dictionary. Thus, the noun will be chosen for *stream.* Furthermore, if the functional frequency of the two categories are on a par, equal weighting will be given to them. Which category will be chosen is determined by the weighting of the phrase structure they are in.

The method of deciding the correct category for a word in the ArchTran system will be improved in the near future using probabilistic model.

In the next section, we will examine the unification operation in more detail from the perspective of sentence processing.

5. Unification Methods

We have already discussed the reason why we use dictionary unification and the dictionary unification principles in ArchTran. In this section we will present possible unification methods. In general, there are three ways to unify the lexical information of identical lexical items in different dictionaries.

[1] **Unifying dictionaries before parsing.** This means that several different dictionaries are unified into one dictionary before any parsing begins. Therefore, only the unified dictionary will be used during dictionary look-up. The advantage of this method is that only one unification action is needed for each lexical item, and thus it saves parsing time. But the shortcoming is that a huge amount of storage space is required for duplicate lexical information in the system, since the merged dictionary and the source dictionaries coexist in the system.

[2] **Unifying dictionaries during parsing.** This means only the lexical items that need to be unified are unified in the course of dictionary look-up and no external dictionary space is needed. The major advantage of this method is the saving of storage space. Nevertheless, this method also has a shortcoming. It requires a special purpose module to handle the unification in the run time and thus increases sentence processing time. Besides, unification has to repeat when a word needs unifying is encountered again.

[3] **Unifying dictionaries with a cache during parsing.** This means cache storage is used to hold the information of the lexical items that are most recently unified. That is, when a word is looked up, the cache will be checked to see whether the word is already there or not. If the word is not in the cache and it is stored in more than one dictionary, all its entries in the various dictionaries will first be unified and then put into the cache. As the cache will be checked when a word is encountered, for a word that is already in the cache, no more unification operation is required next time it is input. This method is similar to that of unifying during parsing, except for the step of checking the cache. The advantage of using cache is that there is no external dictionary space needed and it increases the speed of information retrieval by retrieving information from an internal memory space. But the limitation of using cache is that run-time memory can hold only a limited number of unified words. Another shortcoming is the relative complexity in software, because an additional module has to be added to handle caching.

Comparing the above three methods, we chose to adopt the second method, that is, unifying dictionaries during parsing without a cache. There are three reasons for this decision. First, unlike the use of a merged dictionary, it needs no additional dictionary space. Second, as far as time is concerned, although it requires more time than simply looking up a merged dictionary, the time spent in performing run-time unification is rather small in relation to the

whole MT processing time. Third, it is simpler to implement than using cache and there is no run-time memory limitation problem.

6. Unification Implementation

How does the ArchTran system unify dictionaries? Before answering this question, we will give a brief introduction of the organization of ArchTran.

ArchTran can be decomposed into four general components. **Scanner** looks up dictionaries for the information of lexical items. **Parser** uses the lexical information and analysis grammar rules to analyze the input sentences. And then the **transfer** and **synthesis** modules transfer the English sentence structures into their corresponding Chinese sentence structures. Because the acquisition of lexical information is handled by the scanner, we added the unification module at the scanning stage.

In order to unify dictionaries, there is an interactive user interface **environment control** added to ArchTran, through which user can specify which dictionaries to unify and also specify their hierarchical relation to determine their order in unification. The scanner then looks up the dictionaries specified by the environment control and retrieves the information of lexical items. If there is an identical entry stored in different dictionaries, the scanner calls the unification module to unify the information of the word according to the unification principles.

7. Conclusions

In this paper, we discussed why and how a unification-based lexicography is adopted in the ArchTran English-Chinese machine translation system. Besides satisfactorily handling the problem of different usages or translations of identical terms found in various texts, we also showed that by unifying lexical information in different dictionaries, the time used in dictionary construction is saved; dictionary storage space is minimized; the integrity of distinct dictionaries is preserved; the option regarding which dictionaries to be unified is kept open; and all of the lexical information needed to construct a complete customer-oriented dictionary is available by unifying the relevant dictionaries. Although categorial ambiguity might occur as a result of unification, score function is added as a solution. Unification was proved to be a viable approach for lexicography in an operational MT system.

Using unification in lexicography is one of ArchTran's first attempts to extend the scope of application of the technique. Future research will aim at adopting unification into the ArchTran analysis grammar.

Notes

1. This is the concept of sublanguage-oriented MT system. But the scientific fields to which the ArchTran system is applicable is not so limited as, for example, that of the

METEO system, which aims at translating meteorological reports [Hutc86]. The ArchTran system intends to translate texts of various scientific fields, such as computer, machinery, and so on.

2. From our experience in translating computer articles, it is observed that for different texts of the same domain, the size of vocabulary does not vary to the same extent as the different usages of identical terms. In addition, size of vocabulary is seldom a concern as cost for memory becomes cheaper and cheaper. As for the patterns of syntactic constructions, for sentences within a specific field, they usually do not exhibit an unwieldy variety of structures.

3. The values of the feature SENSE, i.e. SENSEa1, SENSEa2, etc., can be regarded either as semantic types [Alle87] or as any other representation of word sense. It is beyond the scope of this paper to engage in a discussion of word semantics. Besides, for the sake of simplicity, in this example each sense is given a distinct Chinese translation. This ignores the fact that words may be polysemous and therefore more than one sense may be expressed by a single Chinese word. It also ignores the fact that words may be synonymous and therefore more than one Chinese word may be used to express a sense.

References

[Alle87] Allen, J., *Natural Language Understanding*. the Benjamin/Cummings Publishing Company, Inc., U.S.A., 1987.

[Boit87] Boitet, C., *Software and Lingware Engineering in Modern M(A)T Systems*. GETA, University of Grenoble & CNRS, 1987. Prepared for *Handbook of Machine Translation* (Niemeyer 1987).

[Fuji89] Fujitsu. "ATLAS-G for High-level Industrial Document Translation" in *Electronics News from Fujitsu*. Volume 11, No. 3, March 1989, pp. 1-3.

[Hirs87] Hirst, G., *Semantic Interpretation and the Resolution of Ambiguity*. Cambridge University Press, Cambridge, Great Britain, 1987.

[Huan88] Huang, C-R., "Unification" in *Proceedings of R.O.C. Computational Linguistics Workshops I*. pp. 29-54, Academia Sinica, Taipei, Taiwan, 1988.

[Hutc86] Hutchins, W.J., *Machine Translation: Past, Present, Future*. Market Cross House, West Sussex, Great Britain, 1986.

[Nire87] Nirenburg, S., *Machine Translation : Theoretical and Methodological Issues*. Cambridge University Press, Cambridge, Great Britain, 1987.

[Shie86] Shieber, S.M., *Introduction to Unification-based Approaches to Grammar*. Stanford : Center for the Study of Language and Information, U.S.A., 1986.