FAWRMT: WITH SPECIAL EMPHASIS ON GRAMMAR DESIGNS AND PARTITIONED PARSING

Andy Wong Man Hon & Suen Caesar Lun
City Polytechnic of Hong Kong

ABSTRACT

This paper describes the prototypical MT system, FAWRMT: Fully-automated Weather Reporting Machine Translation. The system is not developed just to replicate the Canadian system of TAUM-METEO. Based on the consensus that FAMT is feasible in a restricted domain such as weather reporting, this project also aims at experimenting with corpus-based statistics and analysis, variated grammar designs and partitioned parsing to enhance the efficiency of the system.

1. INTRODUCTION

This project aims at developing an English-to-Chinese/ Cantonese machine translation prototype for Hong Kong weather reporting. The system design is corpus-based and special emphasis has been devoted to the grammar designs and partitioned parsing to cope with the complex text structure in this particular sublanguage. Corpus statistics has been the major focus.

1.1 THE CORPUS

We focused on 31 pieces of weather reports collected from the Hong Kong Royal Observatory in July 1992. Each sample contains 4 sections: (1) General Situation -- condition of the current day, (2) Weather Forecast for Hong Kong -- condition of the following day, (3) Outlook -- condition of the next few days, and (4) Forecast for Macau Today -- current condition in Macau. (Note: A sample report is given in Appendix One)

Below is a statistical summary of the corpus:

Total number of words	=	3,940
Total number of different words	=	493
Average number of words per sample	=	130
Average number of sentences per sample	=	12
Average sentence length (complete sentences)	=	21 wds
Average sentence length (incomplete sentences)	=	8 wds
Average sentence length (overall)	=	14.5 wds

2. SYSTEM OVERVIEW

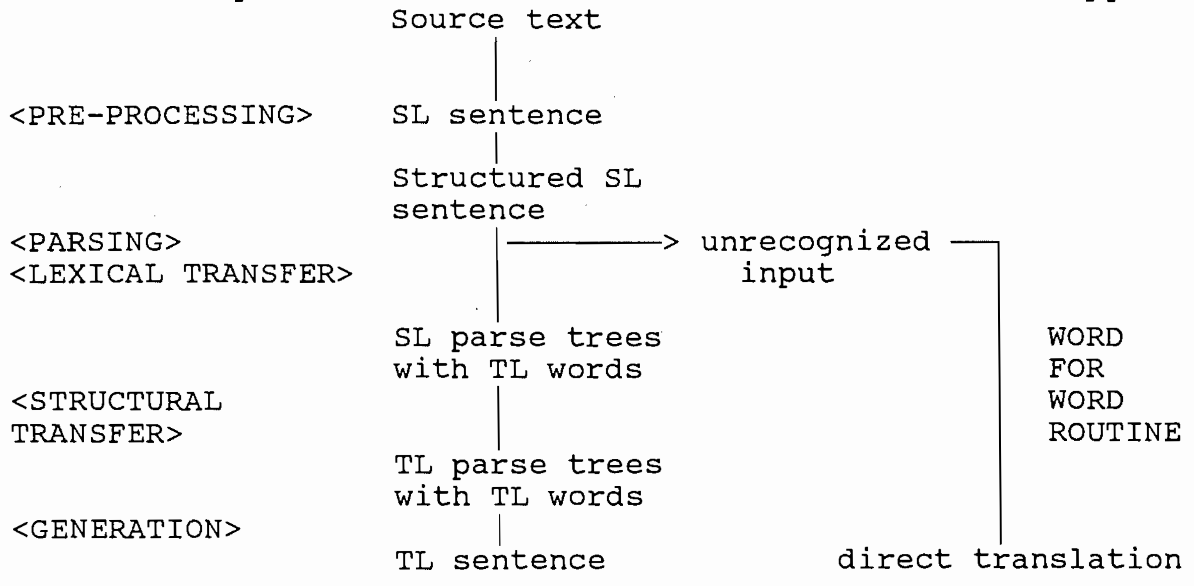
2.1 Basic Characteristics

The system translates single sentences and texts in batch mode with no human intervention. It requires no special formatting on the input apart from the addition of 4 markup symbols to indicate section boundaries, and post-editing is totally eliminated. The program is written in LPA Prolog V2.5 in Eten3/HAN environment, running on IBM pc and accommodating up to 130 words (1 report) per 20 seconds. Up to present it has successfully translated 8 pieces of reports.

To make the system more user-friendly, 2 special facilities are incorporated: (1) double TL Choices: With separate lexicons for Standard Modern Mandarin and Standard Colloquial Cantonese, it allows users to pick the appropriate TL for his own mode of presentation (written or oral). (2) Word-For-Word Routine: The system makes no imprudent guess on unrecognized input elements (wrong spellings / words not found in the lexicons) but triggers this routine to produce a direct translation which suggests what the actual translation should be like.

2.2. Translation Model

We adopt a modified version of the Transfer approach:



(A) PRE-PROCESSING

By identifying typographical features, the system extracts individual sentences from the text, tokenizing each into a machine-readable list form.

(B) ANALYSIS

The parser identifies the structural constituents of the sentence according to the Analysis Grammar, looks up the words in the lexicon, and returns the TL terms and syntactic/semantic information. Finally a SL parse tree is built. The process is syntactically-based, but aided by semantic filtering routines:

LINGUISTIC PRINCIPLES	FUNCTIONS
1. subcategorizations	specifying collocation patterns of verbs and complements in the grammar rules (particularly helpful in handling PP-attachment) e.g. VP --> V_group Complement Complement(iv) --> [] Complement(tv) --> NP

	Complement(dav) --> NP NP (*NOTE: dav - ditransitive verb)
2. selectional restrictions	assigning semantic features to specify what nouns can follow what verbs and can be modified by what adjectives
3. general collocations	specifying whether a verb can be preceded by an auxiliary verb and if so, in what form
4. other syntactic/semantic features	specifying (1) which TL should be selected among homographs, (2) whether an ADJ should be followed by the TL word "de", (3) which classifier accompanies each noun, (4) TL order of adverbs in a compound adverb group

The process runs on a top-down, depth-first and sentence-by-sentence basis (no risk of neglecting contextual factors since intersentencial connection is insignificant in the sublanguage). Morphological treatment is disregarded because of the lack of inflectional variants, while semantic analysis is reduced to applications of semantic features and filtering routines. Since the analysis lexicon and the transfer lexicon are combined into a single bilingual dictionary, Lexical Transfer proceeds along with Lexical Analysis to avoid checking the same lexical entry twice.

(C) TRANSFER

The SL tree is transformed into a TL tree in a top-down, depth-first manner applying transfer rules constructed out of the TL grammar. Lexical routines (short programs) are implemented for special transformations: (1) selection of TL for determiners according to the definiteness of NPs, (2) selection of nominal classifiers, (3) insertion of the TL words "you3" and "de" before nouns and after adjectives wherever appropriate, (4) ordering of adverbs in TL. Finally a TL parse tree is formed.

(D) GENERATION

In the absence of morphological treatment, this module is simplified and mainly involves the decoding of TL trees to extract and arrange words into linear TL sequences.

(E) DIRECT TRANSLATION

If the SL string contains unidentifiable words or phrases, the system will trigger a lexicon lookup routine to produce a direct translation as reference. This prevents over-translation and guarantees the accuracy of output.

3. GRAMMAR TYPES

Two ideas are involved in the grammar design: (1) Multi-Path Grammar, which supports partitioned parsing, and (2) Statistically-Based Grammar, which derives maximum benefit from corpus analysis to facilitate parsing.

3.1 Multi-Path Grammar

A weather report is a mixture of complete sentences, incomplete sentences and domain-specific phrases. A partitioned parser with multiple "grammar paths" is thus used, with a Phrase Structure Grammar Path for parsing complete sentences, a Semantic Grammar Path for parsing incomplete sentences and phrases, and a Heading Path for parsing section headings. The Semantic Grammar comprises the ATM Grammar (atmospheric conditions), TEMP Grammar (temperatures) and WIND Grammar (wind conditions). An automatic recognizer checks the sentence nature (by identifying key word, phrases and structures to trigger the correct grammar paths.

3.1.1 PS Grammars

The grammar is similar to a common one except for including 3 "semantic phrases" to handle domain-specific patterns:

(A) TIME phrases - at sentence beginning for temporal indication:

```
TIME --> [at]  number  [am/pm]  ADVP
```

e.g. <u>At 11 pm last night</u>, tropical depression Dianna was centred about 620 kilometres south of Kagoshima.

(B) SPEED phrases - nominal or verbal PPs describing wind speed:

```
SPEED --> [at/of] modifier number [per] [hour]
```

e.g. (in NP): Strong gusts <u>of around 90 kilometres per hour</u> were recorded at the airport yesterday.

e.g. (in VP): The tropical depression is forecast to move west <u>at around 22 kilometres per hour</u>.

(C) LOCATION phrases - verbal PPs indicating locations of pressure bodies (e.g. depression, trough, ridge...); with 2 components, DISTANCE phrase and DIRECTION phrase:

```
LOCATION  --> DISTANCE  DIRECTION
DISTANCE  --> modifier  number  [kilometres]
DIRECTION --> direction  [of]  proper_noun
```

e.g. Eli was centred <u>about 560 kilometres</u> <u>east of Manila</u>.

3.1.2 Semantic Grammars

The corpus is full of unusual sentence patterns which are totally different from those of ordinary complete sentences, but can be generalized into regular semantic patterns of their own. It is inconvenient and inefficient to treat them with a syntactically-based grammar, but by a "semantic grammar" constructed out of those patterns. "The grammar may have an upper level of semantic categories to mark the semantic patterns, and a lower PS level" (Hutchins 229) to indicate the syntactic constituents in the sentence". This is the case in TAUM-METEO (King 264). The idea is also applied in our system, but has been elaborated, modified and refined to cope with the more complicated sentence patterns. The grammar has 3 parts:

(A) ATM Grammar

These expressions are special combinations of NPs, ADJPs, ADVPS and PPs. The main rule has 4 components:

```
ATM --> COND CMOD TIMES COMPL

COND  (adj/np/vp)  =  the main atmospheric condition
CMOD  (pp)         =  complementary/accompanying condition
TIMES (adv/pp)     =  temporal indication
COMPL (adj/np)     =  complement of the whole ATM phrase
```

Examples:

```
1. Cloudy with scattered showers on Monday.
   <----> <--><-----------------> <------->
    COND   prep      COND           TIMES
           <-------------------->
                   CMOD
   <---------------------------------------->
                   ATM

2. Sunny intervals during the day.
   <--------------> <------------->
        COND            TIMES
   <------------------------------>
                 ATM

3. remaining fine and very hot.
   <--------><---> <-> <------->
      adv     cond conj  COND
   <------------->
        COND
   <---------------------------->
                COND

4. fine at first, becoming cloudy with isolated showers.
   <--> <------> <--------------> <-------------------->
   COND  TIMES        COND               CMOD
                 <------------------------------------->
                                COMPL
```

(B) TEMP Grammar

This is a special case. TEMP expressions are indeed complete sentences. But since they have only 2 patterns, it would be more efficient to treat them as semantic expressions in a "rewriting" manner -- treating the patterns as fixed frames so that the parser will not return the underlying structure of "the minimum temperature is 19 degrees" but will search for "the", "temperature" and "degrees" and pick the correct options from <type> and <verb2>:

```
  [1] Temperatures <verb1>  [num] conj [num] degrees.

      <verb1> =  will range between / will be in the range of/
                 will range from / are expected to range between
  [2] The <type> temperature  <verb2>  [num]  degrees.

      <type>  = maximum/minimum
      <verb2> = will be / will be around / will reach
```

(C) WIND Grammar

WIND expressions are complex NPs reporting wind conditions:

```
WIND  -->  TYPE  DIRECTION  [winds]  COMPL
COMPL -->  MODIFIER  MAIN  PLACE  ADV  TIME
```

TYPE (adj)	=	nature of wind condition
DIRECTION (adj)	=	wind direction
COMPL (adj/vp)	=	complement of the whole WIND phrase
MODIFIER (adv)	=	pre-modifier of the complement
MAIN (adj)	=	main condition of the complement
PLACE (pp/adv)	=	locative indication
ADV (adv)	=	post-modifier of the complement
TIME (adv/pp)	=	temporal indication

Examples:

```
1.  Fresh southwesterly winds.
    <---> <----------->
     TYPE   DIRECTION

2. Light to moderate south to southwesterly winds.
    <---------------> <----------------------->
         TYPE                 DIRECTION
```

```
3.  Moderate easterly winds, occasionally fresh offshore later.
    <-------> <------->      <-----------> <---> <-------> <----->
     TYPE    DIRECTION         MODIFIER    MAIN    PLACE     TIME
                             <---------------------------------->
                                            COMPL

4.  Fresh to strong southeasterly winds, moderating gradually
    <---><--><-----> <------------>      <---------> <-------->
     TYPE conj TYPE   DIRECTION              MAIN        ADV
    <-------------->                     <------------------->
        TYPE                                       COMPL
```

3.2 Statistically-Based Grammar

Grammar clauses are arranged in the program according to their relative frequencies of application. This minimizes back-tracking during run time and enhances processing efficiency.

4. SYSTEM STRUCTURE

4.1. Program Design

The actual program is constructed out of the theoretical translation model. It has the following components:

```
              <LEVEL 1>    <LEVEL 2>            <LEVEL 3>

                           Pre-processor

                           PS Translating        Main Program Body
                           Program               Module 1: ANALYSIS
SYSTEM --> Manager                               Module 2: TRANSFER
           Program                               Module 3: GENERATION
                                                 Chinese Lexicon
                                                 Cantonese Lexicon

                           Semantic Manager      (1) ATM Program
                           Program               (2) TEMP Program
                                                 (3) WIND program
```

The Manager Program takes overall control of the program by identifying the different sections in the text and triggering grammar paths. Actually, Section 1 contains only complete sentences while the other three contain only incomplete (semantic) sentences. This signals when to consult the PS program and the Semantic ones.

Still the system has to select among the three Semantic paths. This is done by the Semantic Manager Program, in which an automatic recognizer looks for key words, phrases and structure to identify the sentence nature, with the following algorithm:

```
INPUT = input sentence

IF INPUT contains the key words "temperature(s)" & "degrees"
in the format [...temperatures ... degrees] THEN goto TEMP

ELSE  IF INPUT contains the word "winds" at the end or at  a
clause boundary THEN goto WIND

ELSE  IF INPUT starts with an ATM-ADJ, ATM-ADV  or  ATM-COND
word (check the ATM Lexicon) THEN goto ATM

ELSE goto PS

Else goto WORD-FOR-WORD
```

The key structures are collected from detailed corpus analysis and are reliable at least in handling the 8 pieces of samples.

The Pre-processor formats each sentence into an analyzable form which then goes through 3 translation modules. The process repeats until all sentences in the section have been translated.

The Manager Program continues to search for the other sections, consulting the corresponding grammar path(s) and translating the sentences until all sections have been processed. The process takes about 20 seconds on average.

Input containing unrecognizable elements can still pass the Pre-processor which only checks typographical features in sentence extraction and tokenization. It is not until the input reaches the parser that it is rejected. This is very important since the unknown elements, though untranslatable, have to be localized into normal tokens so that they can be reported in the the suggested translation produced by the word-for-word routine.

With a pre-processor converting input into machine-readable forms, there need not be any stylistic formatting on the input. Similarly, as the generator decodes all formats and structures in the TL tree, the output text requires no post-editing.

4.2. Lexicon Design

Every translation program has a bilingual lexicon (either the Chinese lexicon or the Cantonese lexicon is consulted once). In the absence of morphological analysis, the lexical entries are simplified. There are no details of agreement, tense, gender, and inflections but only parts of speech, SL and TL terms, and syntactic/semantic information for parsing and transfer.

4.3. Programming Details

4.3.1 Implementation of Rules

Grammar rules and transfer rules are converted to C-Form (Condensed Form) to be processed "deterministically and in real-time" (Krulee 202). A grammar is in C-Form if it is "context free" (Krulee 9) and for each nonterminal, X, there exists at most one rule in the form X --> A,B (A is a terminal, B is anything) and at most one rule X --> e (e is an empty string). For example, the VP rule has multiple patterns which need to be organized in a better way for economy and efficiency:

```
ORIGINAL RULES: vp --> verb np advp
                vp --> verb np pp
                vp --> verb np pp advp ...
                vp --> verb
```

```
RULES IN C-FORM:  vp --> verb M1       M1 --> np M2
                                       M1 --> e
                                       M2 --> advp M3
                                       M2 --> pp M3
                                       M3 --> advp
                                       M3 --> e
```

4.3.2 Modular Programming

The theoretical model of the system is already modular. In implementation, the design is reserved by keeping different modules apart as "black boxes" with no interference in between beside input and output passages. Even the different components in a module are separated in some cases. This enables localization of errors, integration of new rules, strategies and SL-TL pairs (Picken 91). However, LPA Prolog allows only structural but not functional modularity since predicates within a module are not entirely local and invisible in others.

4.3.3 Processing Efficiency

A reliable way to upgrade program efficiency in Prolog is to reduce stack overheads, with first argument indexing -- using the first argument of a predicate as index to distinguish the clause from others instead of keeping the clauses as separate rules. This reduces unnecessary growth of backtracking stack and promotes deterministic parsing. For example, our PS parser contains the following rules:

```
p(1,ADV,PP,P0,P):- advp(A1,P0,P1),p(4,A2,PP,P1,P).
p(1,ADV,PP,P0,P):- pp(PP,vp,P0,P1),p(2,ADV,P1,P).
p(2,ADV,P0,P):- advp(ADV,P0,P).
p(2,[''],P,P).
p(3,AJP,ADV,PP,P0,P):- adjp(AJP,_,P0,P1),p(4,ADV,PP,P1,P).
p(3,adjp(epsilon),[''],pp(epsilon),P,P).
p(4,ADV,PP,P0,P):- pp(PP,vp,P0,P1),advp(ADV,P1,P).
p(_,[''],pp(epsilon),P,P).
```

5. EVALUATION AND DISCUSSION.

1. Grammar Designs

We have adopted 4 approaches in designing the grammars: Statistically-Based Grammar, Multi-Path Grammar, Semantic Grammar and Rewriting-Based Grammar.

(1) Statistically-Based Grammar

This is to take advantage of Prolog's backtracking mechanism. If clauses are arranged out of their relative frequencies of occurrences, the possibility of backtracking will be minimized. Each grammar rule is given a statistical index according to the result of a concordance analysis, which decides their position in the program.

The idea turned out to be workable. An indexed program made fewer backtrackings on average and ran faster. Though the working space saved is small in a single case, the effect is maximized in larger program segments. In the dictionary module, for example, there are hundreds of entries and the effect is more obvious. This is particularly effective in compound-term entries (as the following PN entries taken from the PS Translating Program) where SL terms are implemented as lists which have to be scanned through each time. Obviously the statistical method saves much effort in the long run.

```
dicts(pn,[south,china],TL,[Features]).
dicts(pn,[hong,kong],TL,[Features]).
dicts(pn,[northern,guangdong],TL,[Features]).
```

Nevertheless, the method is sometimes inapplicable:

(1) Recursions:

```
    ADJP --> adj
    ADJP --> adj  ADJP
```

(2) Clauses of extremely low frequency:

	<INDEX>
NOUN --> n	(64.7%)
NOUN --> pn	(22.8%)
NOUN --> n_def	(9.5%)
NOUN --> n_cpd	(2.0%)
NOUN --> pron	(1.0%)
NOUN --> n_dummy	(0.2%)

(3) Optional clauses:

OPTREL --> []	(94.2%)
OPTREL --> [that] VP	(5.8%)

In (1), even if the second clause is used more often, it must not come before the first -- the terminating condition of the recursion. In (2), n_dummy has only one entry but becomes the first clause. The reason is obvious: if the system has to fire this clause, it has to make unnecessary efforts to go through the preceding ones. As we now place n_dummy at the top, even when it is not the one to be chosen, only one backtracking is wasted. In (3), after the empty-list rule is formed into a Prolog clause it becomes an all-pass clause as an "optional" way out in case the main clause does not fit. If it is located at the top, all incoming checks will pass and become meaningless. For these pragmatic reasons, the technique is sometimes inapplicable. But the idea itself is feasible in most cases anyway.

(2) Multi-Path Grammar

It is inefficient to cover the different types of sentences in the corpus with a single parser. Even if such a parser can be produced, it will be clumsy and inefficient. It is also unwise to implement so large a segment in one program module. We therefore partitioned the grammar into separate paths to handle different types of expressions.

Although more programming space is required to implement the grammar so that other program designs must be economical enough to compensate for the loss, the advantages seem to compensate for all that. It solves the problem of inconsistent sentence patterns. While individual grammar paths are small in size, more evaluation space is spared which ensures faster processing. It also supports modularity through the division of labour.

The value of sublanguage in MT is that the domain-specific patterns bring great convenience in building the grammar. If we use a single PS Grammar and arbitrarily fix all patterns under the traditional PS categories, we will possibly ignore their semantic natures. The parse tree produced will also be unable to reflect the sentence content.

(3) Semantic Grammars

To utilize the domain-specific patterns, we may either reserve the use of PS grammar but accommodate the rules to the patterns, or generalize the semantic natures of patterns into a semantic grammar. In the latter case, a completely new grammar is produced which analyzes a sentence in a slot-filling manner. See how the following ATM sentence is analyzed in both ways:

[Sunny with some showers in the morning, hot in the afternoon.]

```
PS:  SENTENCE --> ADJP PP PP ADJP PP     ADJP --> adj
                                         PP --> prep NP
                                         NP --> det n
```

```
Semantic:   SENTENCE --> COND CMOD TIME COMPL

            COND --> sunny
            CMOD --> with some showers
            TIME --> in the morning
            COMPL  -->  hot  in  the  afternoon
```

Obviously the modified PS grammar is clumsy and hard to comprehend, nor does it accurately reflect the functional and structural relationships between the constituents. If the pattern occurs regularly in the corpus and has its own semantic structure, why not generalize it into a semantic grammar? Actually, the semantic grammar gives a more reasonable representation of the constituents in a general way, representing a universal framework. Sentences fit into it with their sub-parts filling in the appropriate slots. While the grammar is specifically designed and corpus-based, it has no unnecessary element. It also does not stick to any traditional framework and can be flexibly reformed and extended to cope with problems in parsing so as to resolve structural ambiguities. More importantly, the grammar, enclosing a patterning framework of the sublanguage, becomes a special type of "knowledge representation based on semantic primitives" (Nirenburg 31), and is surely an economical one since the grammar and the knowledge schema are combined into one!

At first sight the idea runs the risks of lacking generality and poor syntactic indication since the semantic types may have no straight forward correspondence with the syntactic structures. It also takes extra time to learn. But while the grammar is specific to a subworld, the rules need not be so general. Furthermore, the grammar can include syntactic categories at a lower level to represent the syntactic nature (Hutchins 229).

<u>(4) Rewriting-Based Grammar</u>

The TEMP program is the shortest and runs fastest among the four, implying that this grammar saves programming space and enhances processing speed. The only disadvantage remains the inability to reveal the syntactic structure of the sentences. But while the sentences covered by the grammar appear regularly, their structures should be familiar. Is there any need to check the structures over and over again?

5.2. Problems and Solutions

During the system development, various problems have arisen, and have been tackled completely or partially in different ways.

(1) Ambiguities

Owing to the restricted vocabulary in the sublanguage, lexical ambiguity is rare and arises only when there are several TLs for the same word -- translational ambiguity (King 262). Three solutions have been adopted:

(A) SELECTIONAL RESTRICTIONS:

e.g. "rain" has different TLs in the following contexts:

1. Rain became much less frequent and intense yesterday.
2. More than 70 millimetres of rain fell over the territory.

WORD	TL	SEMANTIC FEATURE
rain[1]	yu3 shi4	[RAIN]
rain[2]	yu3 liang4	[QUANT]

QUANT phrases must take a [QUANT] noun, so rain[2] is chosen in sentence 2. The mismatch between [QUANT] and the context of

Sentence 1 forces the selection of the [RAIN] entry.

(B) FUNCTIONAL CATEGORIZATIONS: Some word classes are highly categorized according to their structural roles so that correct TLs can be chosen in different contexts:

```
    conjunctions --> pre-sentence conj, inter-sentence conj,
                     intra-np/vp conj, inter-vp conj
    nouns        --> n, proper n, compound n, definite n
    verbs        --> v, phrasal verb
```

(C) FLEXIBLE GRAMMAR DESIGN: Unlike PS Grammar, Semantic Grammars do not adhere to any prescriptive framework and can be easily reformed for disambiguation.

Despite their abundance in the corpus, PPs are restricted to locative phrases where the prepositions have constant senses. Moreover, the SL always has null TL. Therefore prepositional ambiguity is negligible. In addition, while pronouns are rare in the corpus, referential ambiguity is also negligible.

(2) PP-Attachment Problems (Structural Ambiguity)

The problem rests on noun PPs and verb PPs, mainly in transitive VPs. e.g.

[1] An active trough of low pressure brought disturbed weather <u>to the coastal areas</u> <u>of Guangdong</u> yesterday.

[2] Afternoon temperature in the region of 30 degrees were recorded over various parts <u>of the territory</u>.

In [1], it is unclear whether the to-phrase modifies the noun "weather" or the verb "brought". In [2], it is unclear whether the of-phrase accompanies the noun "parts" or the verb "recorded". To tackle this problem, three methods have been used:

(a) Subcategorization - "brought" is marked as a ditransitive verb which must occur in the pattern " brought X to Y".

(b) Modeling common usage - preferences are assigned to the prepositions to mark whether they usually follow noun or verb. Since "of" has the [np] feature, it is believed to follow the noun.

(3) Semantic Grammars: The flexibility of semantic grammar enables it to be easily modified. PPs can be categorized into separate semantic classes or phrases according to their functions so that they will not "clash" together to cause ambiguity. But drawing semantic grammars from complete sentences is complicated, requiring detailed analysis of the sentence structure.

As PP-attachment problem is not serious in the corpus (since most verbs are intransitive and most PPs are sentence PPs), these methods are good enough for disambiguation. But method (b) is ad hoc and insecure. Method (a), though more general, may be risky -- even if a verb has no PP complement, it can have any number of PP adjuncts! So other methods must be used to handle the problem on further extensions.

(3) Verb-Based Transformation

Some of the complete sentences require special transfer formats. So we determine the transfer pattern by the verb type by classifying the verbs according to how they affect structural transfer: TYPE 1 - common verbs, TYPE 2 - passive voice, TYPE 3- "expected" in "is expected to" , TYPE 4 - ditransitive verbs.

(4) Pragmatic considerations

Despite the absence of formal pragmatic processing, there is a mechanism checking the referencing status of the definite determiner "the" for picking correct TLs: nouns with exophoric referents (e.g. the coast, the airport) are classified as "definite nouns". If a noun has a modifying PP and is neither a proper noun nor a "definite noun", it must have no preceding referent since it has to be specified by the PP. On the contrary, a definite noun, though not mentioned previously and not accompanied by a pp, is assumed to have exophoric referents and is treated as old information.

```
e.g. 1. Some thundery showers brought nearly 40 millimetres of
        rainfall to that region.
        (old information - has no post-modifying PP
                         - is not a pn or n_def)

     2. A monsoon prevailed over the coastal areas of Guangdong.
        (new information - has post-modifying PP for specifying
                         - is not a pn or n_def

    *3. Pressure is low over the Western Pacific to the east of
        the Philippines.
        (old information - it is a n_def, so the following PP is
                           not restrictively specifying it!)

     4. Strong gusts were recorded at the airport.
        (old information - it is a n_def)

     5. Showers continued to affect the territory yesterday.
        (old information - it is a proper noun)
```

(4) Adj-PP attachment problem

ADJs and PPs are both pre-modifiers of NP in the TL. How should they be ordered? Results of corpus analysis reveal that PPs containing a proper noun come before ADJs, or else after it.

(5) Adv-attachment problem

Whether an adverb modifies the whole VP or the verb itself determines the transfer formats. This ambiguity is tackled by isolating the 2 types and checking whether there is a verb-modifying adverb immediately after the main verb.

(6) Adv-grouping problem

In a compound adverb phrase, the SL order of ADVs may not be the same as the TL one. e.g.

```
Eng: early yesterday morning
       1       2         3

Chi: zuo2tian1 zao3shang chu1qi1
         2        3         1
```

So semantic features are used to mark the ordering preference:

1.	[manner]	: gradually,generally,much
2.	[place]	: locally
3.	[t1]	: yesterday,today
4.	[t2]	: morning,afternoon
5.	[t3]	: early

(*NOTE: 1-> 5 decreasing preference)

5.3. Translation Quality

The translation quality is satisfactory except in 2 cases: (1) unusual, especially literary, styles are used, (2) while English is subject-prominent and Chinese is topic-prominent, there is sometimes a strong discrepancy in the topic-comment patterns of the SL and TL.

e.g. Some morning showers brought more than 10 millimetres of rainfall to the territory.

MT: Ji3zhen4 zao3shangde zhou4yu3 wei4 Ben3gang3 dai4lai2le shi2 hao2mi3 de yu3liang4.

Human: Zao3shang you3 ji3zhen4 zhou4yu3, wei4 Ben3gang3 dai4lai2le shi4 hao2mi3 de yu3liang4.

The SL and TL topics are "showers" (zhou4yu3) and "morning" (zao3shang) respectively. With the absence of topic analysis in the system, the difference cannot be identified so that "showers" remains as the topic and "morning" remains as the adjective in the MT version -- resulting in Europeanization. On further extension, a topic analysis component can be added to tackle the problem, which will require a knowledge base to incorporate world knowledge and Chinese linguistic knowledge to detect topic conversion between English and Chinese.

5.4. FUTURE DEVELOPMENT

There can be further utilizations of the semantic patterns by going beyond the sentence level to discourse -- with a Text Grammar (Grishman, Kittredge 138). The idea is to represent discourse patterns in the grammar and parse the passage as a single unit, which enables more systematic analysis of the text. This is especially useful since we depend on the textual organization of sections in triggering grammar paths.

Grammar rules are now implemented as Prolog clauses and fired directly. If the system is to be enlarged, we had better separate them from the processing algorithms as data, so that new rules can be integrated independently (Picken 85).

Assistant facilities such as a lexicon updating component can be integrated. There can also be a dialog manager which asks the user to update unknown words interactively, which promotes better machine-human cooperation in producing better results.

6. CONCLUSION

The project marks another attempt of fully automatic translating in a restricted domain. The system produced is experiental but operational. New grammars, partitioned parsing, and various methods were put forward as an attempt to improve processing efficiency. Moreover, the sublanguage model constructed leaves a hint to future research in the same domain.

REFERENCES

Allen, James. Natural Language Understanding. London: Benjamin Cummins Publishing Company, 1987.

Bourbeau, Laurent and John Legrberger. Machine Translation. Philadelphia: John Benjamins, 1988.

Chen, Hsin-Hsi and Chen Kuang-Hua. "Attachment and Transfer of Prepostional Phrases with Constraint Propagation." Computer Processing of Chinese and Oriental Lanugages. 6.2 (1992): 123-142.

Grishman, Ralph and Richard Kittredge. Analyzing Language in Restricted Domains: Sublanguage Description and Processing. London: Lawrence Erlbaum Associateds, 1986.

Hutchins, W. J. Machine Translation: Past, Present, Future. Chichester: Ellis Horwood, 1986.

Kay, Martin. "Machine Translation." American Journal of Computational Linguistics. 8.2 (1982): 74-78.

King, Margaret. Machine Translation Today. Edinburgh: Edinburgh UP, 1987.

Kittredge, Richard and John Lehrberger. Sublanguage: Study of Language in Restricted Semantic Domains. New York: Walter de Gruyter, 1982.

Krulee, Gelbert.K. Computer Processing of Natural Language. London: Prentice Hall, 1991.

Lauren, Christer and Marianne Nordman. Special Language: From Human Thinking to Thinking Machines. Britain: Multilingual Matters, 1989.

Nirenburg, Sergei. Machine Translation: Theoretical and Methodological Issues. Cambridge: CUP, 1987.

Pereira and Shieber. Prolog and Natural Language Analysis. Menlo Park: CSLI, 1987.

Picken, Catriona. Translating and The Computer 8: A Profession on the Move. London: Aslib, 1987.

Smith, George W. Computer and Human Language. New York: OUP,1991.

Steel, Brian D. LPA Prolog Professional Compiler V2.5. London: Logic Programming Associates, 1988.

Williams, Stephanie. Humans and Machine. Norwood: Ablex Publishing Corporation, 1985.

天氣概況：

尋日朝頭早幾陣局部地區性雷雨影響本港，但係雲層逐漸變得輕薄，下晝帶嚟晴朗嘅天氣。喺西太平洋，尋日有一股熱帶低氣壓增強咗成為一股強烈嘅熱帶風暴，個名叫做艾里。 尋晚11點，艾里喺馬尼拉以東大約560公里結集，估計會增強，以每小時大約30公里嘅速度向西北偏西移動。估計艾里會迅速橫過菲律賓，禮拜日進入南中國海。

本地天氣預測：（11/7/1992　禮拜六）

天晴。
氣溫喺26同31度之間。
吹和緩嘅東南風。

禮拜日天氣展望：

初時天晴，晏啲漸漸會好大風，有幾陣雨。

澳門地區今日天氣預測：

多雲，有局部地區性嘅驟雨，日間間中會有陽光。
氣溫喺25同31度之間。
吹輕微至和緩嘅東至東南風。

```
C:\>_
```

天氣概況：

尋日下晝初時幾陣局部地區性同埋強烈嘅雷雨影響西貢，為嗰個地區帶嚟咗大約40毫米嘅雨量。而家，有一道高壓脊由太平洋向西面移動，估計禮拜六會影響廣東沿岸地區。尋晚有一個西太平洋低壓區增強咗成為一股熱帶低氣壓。尋晚11點，喺馬尼拉以東大約1240公里結集，估計會以每小時大約22公里嘅速度向西面移動。

本地天氣預測：（10/7/1992　禮拜五）

除咗有幾陣驟雨，大致上都係天晴。
氣溫喺26同31度之間。
吹和緩嘅南至東南風。

禮拜日天氣展望：

禮拜六天晴同埋炎熱。

澳門地區今日天氣預測：

初時大致上都係多雲，有局部地區性嘅驟雨，晏啲漸漸會天晴。
氣溫喺27同32度之間。
吹輕微至和緩嘅南至西南風。

```
C:\>_
```