Bi-Lingual Sentence Generation*

Chung-Cherng Chen and Jyun-sheng Chang
Institute of Computer Science
National Tsing Hua University

Abstract

In this paper, we use a single internal representation for *bi-lingual* document generation (English and Chinese) to experiment on the feasibility of a language independent structure. The input is designed based on the *case relation* and the sentence generator is implemented using *systemic grammar*. We augment the systemic grammar with *procedural attachment* to deal with language dependent choices. The system is tested for the application domain of preparing technical manuals with satisfactory results.

1. Introduction

The aim of this paper is to experiment on bilingual generation of technical document.

The input is in some language independent internal representation and the output is sentences in a target language. There are two important issues in this process. One is the structure of internal representation and the other is the design of sentence generator. We will discuss these two issues in the following sections.

---

* This research is partially supported by National Science Council, grant No. NSC80-0408-E007-13.

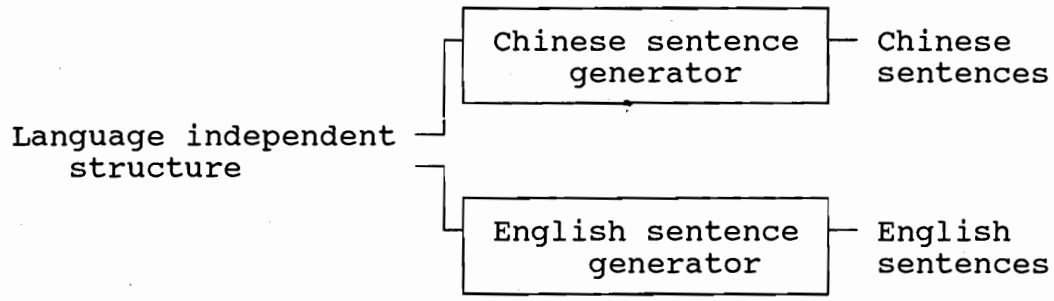

Figure 1. A Bi-lingual Sentence Generator

There are many advantages in using language independent structure in sentence generation. For example, in machine translation, the language-independent structure can be used as the interlingua so that an MT system decomposes into two modules, one for parsing and one for generation. Both modules can be used for translation of other language pairs with the same source or target language. For a text generation system, a knowledge base that yields language-independent structure can be used to generate multilingual document.

A sentence generator usually is based on some kind of grammactical formalism. Several grammactical formalisms have been used for sentence generation, including ATN [Goldman 1975], systemic grammar [Davey 1975, Mann 1973, Patten 1985 and Kuo 1989], and transformational grammar [Mckeown 1979 and Maudlin 1984]. All of these systems deal with one specific target language such as English and Chinese.

This paper continues previous work on Chinese sentence generation [Kuo 1989] and focuses on the following problems:

1. Making the representation more language independent.
2. Extending the scope of sentence generation.

3 Using a single internal representation for bilingual document generation to illustrate the feasibility of a language-independent representation.

This section introduces the problem that this paper sets out to solve and previous work on sentence generation. Section 2 presents detail description of the language-independent structure. Section 3 describes both the Chinese and English sentence generators. Section 4 presents examples to illustrate the entire process of generation; the differences between Chinese and English will be emphasized. Section 5 concludes the paper with a few remarks.

2. Internal Representation

The Chinese sentence generator described in [Kuo 1989] is not very language-independent. It has the following shortcomings.

1. The internal representation is insufficient to represent the events Considering the following two sentences

   張三用鑰匙打開門
   鑰匙打 開門

   The input for sentence (1) is

   Agent : 張三

   Adv : 用鑰匙

   The input for sentence (2) is

   Agent : 鑰匙

" 鑰匙 " plays the same roles but has different input forms in the above two sentences.

2. Users must specify some features particular to the target language. For example, in certain situation, users must specify *ba or passive* which has nothing to do with the semantics. We'll overcome this shortcoming with *procedural attachment.*
3. Considering the following two sentences

   他放一本書在桌子上

   他在桌子上睡覺

   The location in the above sentences has different surface structures. If we are not careful, we'll generate an illegitimate sentence such as

   * 他睡覺在桌子上

Removing these shortcomings and making the sentence generator language independent are the goals of this paper. The following sections will show how the problems are resolved.

In the following, we will describe the internal representation used in our sentence generator. The goal of the internal representation is to record the meaning of a sentence. A sentence with more than one meaning should have more than one internal representation. Similarly, sentences with different syntactic structures but the same meaning should get mapped to the same structure.

We adopt case grammar as the basis of our internal representation for the following two reasons:

1 It is fundamentally a theory of meaning.

2 The number of possible cases is quite small, so it is easy to manipulate and to use in internal representation.

Case Grammar and Case Relation

The essence of case grammar is that the semantics of a sentence can be expressed in terms of case relation which is the relationship between noun phrases and the verb. Typical cases are as follows.

1 *Agent* : A noun phrase fills the agent case if it describes the instigator of the action described by the sentence. For example,

John broke the window.

2 *Theme* : An NP that describes something undergoing some change or being acted upon will fill the theme case. For example,

John broke the rock.

3. *Instrument* : An NP is an instrument if it describes a tool, material or force used to perform some event. For example,

Jack saw the ship with the telescope.

4 *Experiencer* : An animate entity is an experiencer if the entity is in a desired psychological state or undergoes some psychological process such as perception. For example,

John saw the unicorn.

5 *Beneficiary* : The case is filled by the animate person for whom a certain event is performed, as in

I gave the book to Jack for Susan.

6 *At-Loc* : This case describes the location where the action happened as in

He sat on the chair.

7 *From* : The case can be further divided into two subcases *from-loc* and *from-poss*, representing the source of a movement action and the original owner respectively.

8 *To* : The case can be further divided into two subcases *to-loc* and *to-poss*, which mean the destination and the new owner respectively.

9 *Time* : This case indicate the time when the action happened.

10 *Direction* indicates the direction of the action.

Case grammar can be stated in the following three rules.

1 Sentence --> Modality + Proposition

2 Proposition --> V + C_1 + C_2 + ... + C_i + ...+ C_n

3 C_i --> K + NP

The first rule indicates that a sentence consists of *modality* and *proposition*. Proposition is a tenseless set of case relations. *Modality* concerns about *mood* information such as *indicative*, *imperative* and *tense* information such as *present* or *past*.

The third rule states that each case consists of a case marker and a noun phrase. The case marker will be realized as a preposition ('in' and 'at' in English, or '把' and '在' in Chinese) or by the position in the surface structure. For example the *Agent* case is usually realized by the *subject* position in the surface structure.

The structure of Verbs

Some cases are intimately related to the verbs. We call them inner cases. Other cases are optional (called outer cases). There are two syntactical property associated with an inner case:

1 An inner case must always appear, For example the verb坐has the inner cases *Agent* and *Location*.

他坐在椅子上

2 There are different kinds of case marker associated with an inner case. For example,

老師坐在講臺上 .....normal order

講臺上坐著一位老師 .....locative inversion

老師在講臺上坐著 .....locative preposing

Internal Representation

According to case grammar, we classify the internal representation into three categories: *events, entities*, and *predicate*s. Events concern about something that happened. It consists of *modality information* and *case information*. Case information specifies the kinds of information about this event. There are two kinds of case information. They are *semantic cases* and *discourse cases*. Semantic cases express the semantic meaning and the discourse cases conveys the discourse information such as *focus* and *understanding*. The discourse cases affects the syntactical choice and the surface structure of a sentence. Semantic cases can be further divided into two kinds. One is the *basic case* which we discuss in 3.1.1. The other is the augmented case which we used to express more complicated meanings. We added augmented

cases to generate bounded sentences while maintaining generality. The following augmented cases are allowed:

1. *Time* : indicating that the subordinate event happened either before, after or at the time of the main event. For example,

   After you decided on a location for your printer, the first step in setting it up is to install the paper feed knob.

2 *Result* : indicating the resultant event of the main event. For example,

   Save it so that we can move the printer with it when we move the printer.

3 *Purpose* : indicating the purpose of the main event.

4 *And-then* : indicating the subsequent event of the main event.

5 *According* : indicating the method of the main event.

These three kinds of case are illustrated in Figure 2 through 4.

The corresponding linear representation is

```
( EventName    ( Event ( Modality Information )
                 ( Agent ( Entity .....))
                 ( Theme ( Entity .....))
                 ( Pred ( Predicate .....))
                 ( Experiencer ( Entity .....))
                 ( At-Loc ( Entity .....))
                 ( To-Loc ( Entity .....))
                 ( To-poss ( Entity .....))
                 ( From-Loc ( Entity .....))
                 ( From-poss ( Entity .....))
                 ( Instrument ( Entity .....))
                 ( Time ( Entity .....))
                 ( Direction ( mod .....))
               )
)
```

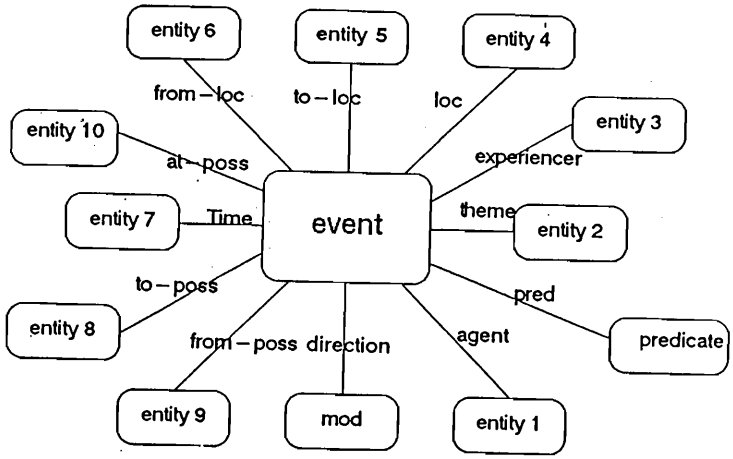

Figure 2. Basic Cases

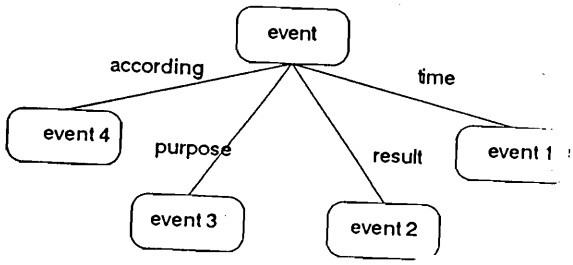

Figure 3. Augmented Cases

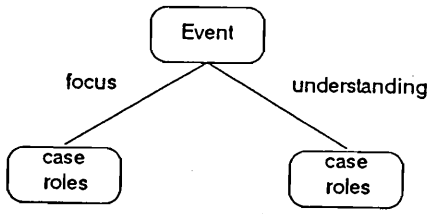

Figure 4. Discourse Cases

Entities can assume a case role such as *agent, location, theme* and *time* and is represented by a head noun, definite/nondefinite information and modifiers such as *adjective, location* and *genitive*. See Figure 5.

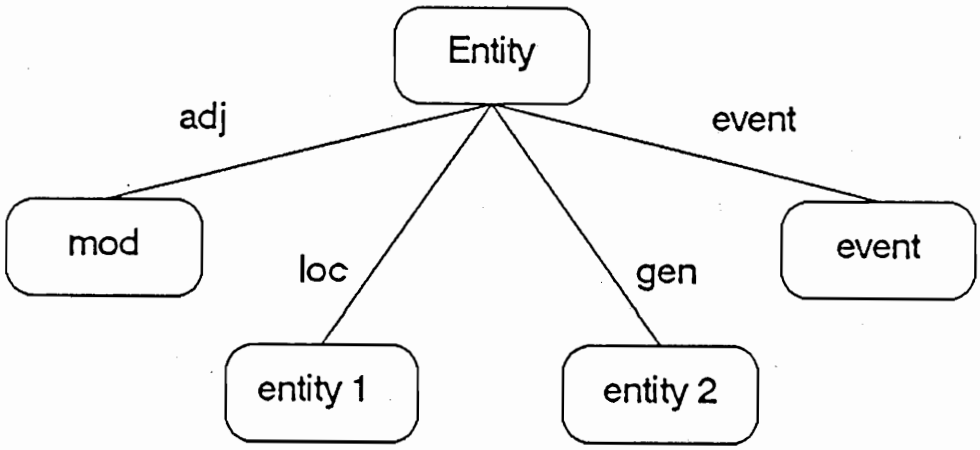

Figure 5. The Representation for an Entity

The corresponding representation is

```
( CaseName      ( Entity Singular/Plural Number Defnite/Nondefinite )
                (Loc ( Entity . . . . . .))
                (Gen ( Entity . . . . . .))
                (Event  ( Event . . . . . .))
                (Adj ( Mod . . . . . .))
```

The *Predicate* case represents the kind of action involved in an event. It may be accompanied by modifiers (adverb).

An Example

For example, a sentence from Epson Printer User Manual

Hold both ends of the tracter unit and slowly tilt the unit.

has the semantic representation as shown in Figure 6. And the semantic net can be linearized as follows.

```
(event1     (event imperative present)
            (pred(action (verb hold)))
            (Theme      (entity definite third plural (hn end))
                        (adj ( mod (mod both)))
                        (gen ( entity definite third singular (hn tracter))))
            (and-then   (event present )
                        (agent (entity definite second singular pronoun (hn you)))
                        (pred (action (verb tilt))
                              (modifier (mod (mod slow))))
                        (Theme (entity definite third singular pronoun (hn unit)))))
```

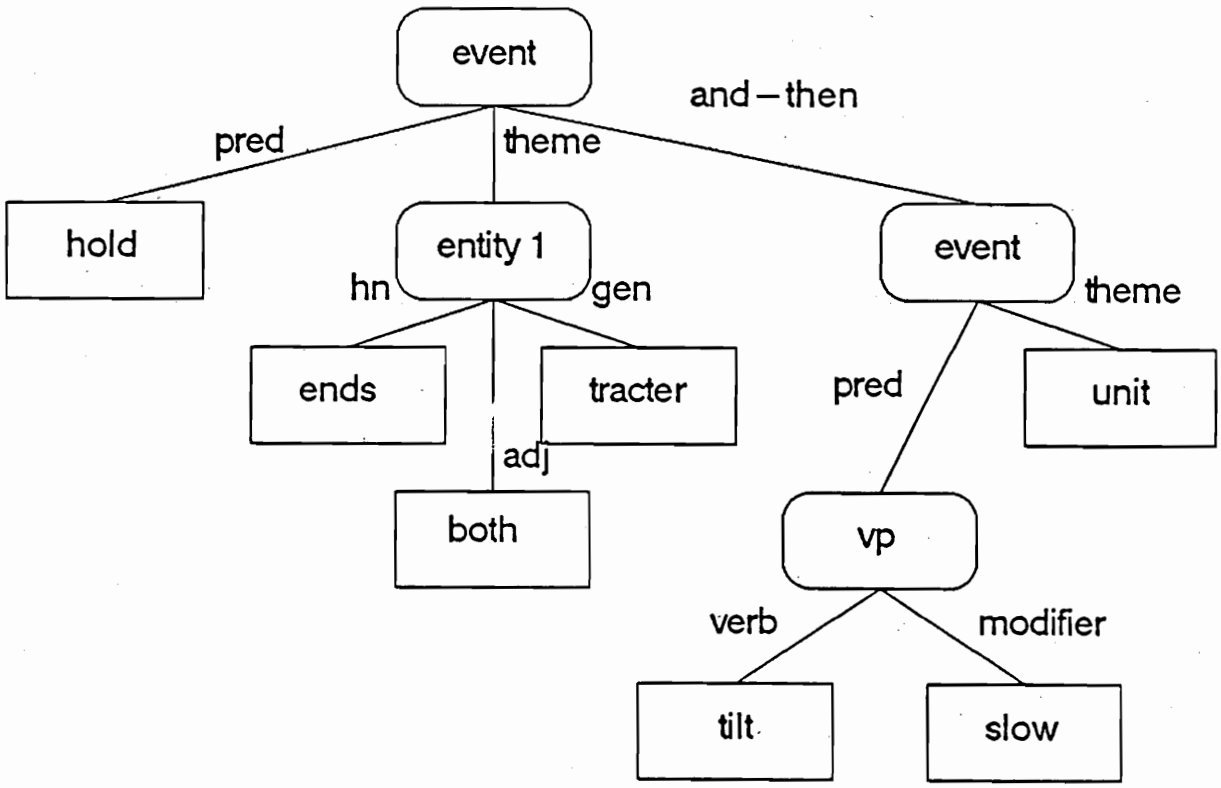

Figure 6. Representing "Hold both ends of the tractor unit and slowly tilt the unit"

3. Sentence Generation

The sentence generator accepts the *semantic case*s and *discourse case*s to generate an adequate sentence. The major mechanism to make syntactical choice is *procedural attachment*. In this paper, we'll show how the sentence generator manipulate these two kinds of cases through attached procedures.

The realization rules in the original systemic network are not sufficient for text generation in English and Chinese deterministically. We have proposed to add some notation to resolve the problems.

The ? Notation

While generating English sentences, we must enforce the *subject verb agreement*. So we need to know the number and singular/plural information about the subject. Here, we define a notation :

?(element feature)	to mean	whether the element has the feature
?('$ feature)	to mean	whether the current element has the feature
?(element 'self)	to mean	whether the element exists

For example *?('agent 'plural)* means whether agent has the *plural* feature; *?('$ 'present)* means whether the current clause has the present feature; *?('agent 'self)* means whether the agent exists.

The # Notation

During the generation of event, we should know whether the event is dependent or independent, whether it is a bounded or relative clause. This information must be obtained from the upper level. So we need a realization rule for *inserting feature*. We

define realization rules (# *element feature-list*) as inserting *element* to *feature-list*. For example, when we generate the time- event, we have the realization of (# time (dependent bound time-bound)) which means the clause we generate next will have the feature *dependent, bound, time-bound.*

The Grammar

According to the internal representation, we classify the network into three levels : *event, object, predicate*. Each level of representation is handled by the corresponding network. We will sketch the grammar for both Chinese and English in the following. Refer to [Chen 1990] for more detailed description.

The Action-type System (Chinese)

The *action-type system* processes the action of the event. The philosophy that we take in action-type system is the **classification of verbs**. By classifying the verbs, we know the inner cases and the corresponding syntactical choice. We can manipulate the inner case and choose the adequate sentences according to the discourse function. The action-type system consists of four subsystems, and we will illustrate only the action and aux subsystems in the following.

The Action System (Chinese)

The action system distinguishes a *tran-action (transitive)* system from an *intran-action (intransitive)* system. The *intran-action* system consists of *vocal intran-action-location* and *intran-locomotion* system.

The *intran-location* system can have three syntactical choices, *normal-order, locative-inversion and locative-preposing*. For example

老師坐在講臺上　　normal order

講臺上坐著一位老師　　locative inversion

老師在講臺上坐著　　locative preposing

The condition for the locative inversion is *agent nondefinite* and *loc definite*. This means that agent is the new information and location is the old information. It is often called *presentative clause*.

The condition for locative preposing is *agent* and *loc definite*. We realize these conditions by attaching a procedure to the system of *intran-location*. The following is the corresponding code:

```
( cond      ( ( and ?( 'Agent 'Definite ) ?( 'Loc 'Definite ))    '( Locative-Preposing
))
            ( ( and ?( 'Agent 'Nondefinite ) ?('Loc 'Definite )) '(          Locative-
Inversion))
            ( T                                                   '( Normal-Order )))
```

Similarly, the *Tran-Action* consists of *Regular* and *Tran-Location* system. The *regular* system has the following four syntactical choices.

normal :

老闆開除了他

topic :

condition : Theme is definite

他老闆開除了

preverbal :

condition : 1 Theme is definite

2 Verbs contains disposal feature.

老闆把他開除了

passive :

condition : Focus is Theme

他被老闆開除了

and the procedure attached is:

```
( cond      ( ?( 'Focus 'Theme )                              '( passive ) )
            (( and ?( 'Theme 'Definite ) ?( '$ 'Disposal ))   '( preverbal)
            ( T                                               '( Normal )))
```

Similar work is done for the *tran-location system*.

The Event Network (English)

One of the main differences between the English network and the Chinese network is *voice* and *subject verb agreement*. The voice system will select the active or passive system according to *focus*. The *active* system consists of *select-subj* and *theme-obj* systems. The select-subj system will select the *subject* from *agent, experiencer* and *instrument*. If the *subject* is *the agent*, then there are two systems included: *agentnn and agentqq*. Agentnn has three alternatives, *subj-first, subj-second* and *subj-third*. We attach to agent-nn a procedure to choose one from the alternatives.

```
( cond      ( ?('agent  'first )      '( subj-first ) )
            ( ?('agent  'second )     '(subj-second ) )
            ( ?('agent  'third)       '(subj-third)))
```

Similar work is done for agent-qq which manipulates the singular/plural features.

4. Examples

In the following, we show a list of sentences used in the experiment. The sentences are listed in the sequence of (1) an original Chinese sentence (2) the generated Chinese sentence (3) the original English sentence (4) the generated English sentence.

1.1 當 你 打開 印表機 的 包裝材料 時 請依 下 圖 所示檢查 是否 所有 的配件 齊全了

1.2 當 你 打開 印表機 的 包裝材料 時 檢查 是否 你 有 下 圖 顯示 的 所有 配件

1.3 After you unpack the printer, check that you have all parts shown below .

1.4 when you open the packaging materials of the printer check whether you have all parts that the following figure shows

2.1 拿出 配件 後 將 包裝材料 保存 以便 將來 搬運 印表機 時 用 得 著

2.2 當 我們 拿掉 配件 以後 保存 包裝材料 使得 將來 我們 能 用 它 搬運 印表機

2.3 After removing the parts, store the packaging materials in case you ever need to transport your printe.

2.4 after we remove the PARTS store the packaging materials so that we can transport the printer with it later

3.1 拿掉 拖曳式 牽引器

3.2 拿掉 拖曳式 牽引器

3.3 Remove the pull tractor

3.4 remove the pull tractor

4.1 為了便利 自我測試功能 的執行 拿掉 拖曳式 牽引器

4.2 拿掉 拖曳式 牽引器 使得 將來 我們 能 執行 自我測試

4.3 In preparation for performing the self test later, remove the tractor.

4.4 remove the pull tractor so that we can perform the self test later

5.1 將牽引器 的 蓋子 直立 然後 向上 提起

5.2 將 牽引器 的 蓋子 直立 然後 將 它 向上 提起

5.3 Raise the tractor cover then lift it up.

5.4 raise the cover of a tractor and-then lift it up

6.1 拿掉 插 在 牽引器 的 兩 端 的 包裝材料

6.2 拿掉 插 在 牽引器 的 兩 端 的 包裝材料

6.3 Remove the packaging materials inserted between both sides of the tractor

6.4 remove the packaging materials that•is inserted between both sides of the tractor

7.1 保存 它 以便 再 搬運 印表機時 用得著

7.2 保存 它 使得 當 我們 搬運 印表機 時 我們 能 用 它 搬運 印表機

7.3 Store it in case you ever need to transport the printer.

7.4 store it so that when we transport the printer we can move the printer with it

8.1 握住 牽引器 的 兩 端

8.2 握住 牽引器 的 兩 端

8.3 Hold both ends of the tractor.

8.4 hold both ends of the tractor

9.1 慢慢地 將 它 向後 頃斜

9.2 將 它 向後 慢慢 頃斜

9.3 Slowly tilt it back

9.4 slowly tilt it back

10.1 牽引器 的 缺口 會 從 扣針 鬆開

10.2 牽引器 的 缺口 會 從 扣針 鬆開

10.3 The notches of the tractor will snap free from the mounting pins

10.4 the notches of the tractor will snap free from the mounting pins

11.1 把 牽引器 向上 提起

11.2 將 牽引器 向上 提起

11.3 Lift the tractor up

11.4 lift the tractor up

5. Conclusions

In this paper, we focus on the topic of language independent. We adopt the case relation to be our input and we use procedural attachment to deal with the language dependent choice. We implement both the English and Chinese Systemic Grammars to demonstrate that our input is in some way language independent.

The domain that we test our system is the technical manual, Epson user's guide for the LQ-500 printer.

Our system can be improved in the following respect:

(1) Extending the *scope* of the grammar : We implement just a subset of English and Chinese grammar. There are still many grammatical phenomena we didn't implement such as progressive, perfect tense.

(2) Improving the *procedural attachment* mechanism: We can improve the procedural attachment mechanism to make the sentences generated more elegant. For example, We can realize the focus of event using *cleft* sentences instead of using only *passive* sentences in Chinese sentence generation.

(3) Adding a *preprocessor*: Consider the sentences below.

Unpack the printer.

打開 印表機 的 包裝材料

In order to generate this pair of sentences, we will need a preprocessor between the input and the sentence generator to deal with difference in lexical expression in different target language.

(4) Our system now generates a clause for the internal representation of an *event*. We can add variety to the text by generating a different kind of grammatical

constituent. For example, we can generate with a noun phrase in a prepositional phrase for a *result-event* as in Sentence 4.3.

References

[Chen 1990]

C.C. Chen, *A Multi-Lingual Sentence Generator Based on Systemic Grammar*, Master Thesis, Department of Computer Science, National Tsing Hua University, Hsinchu, Taiwan, 1990.

[Davey 1978]

A. Davey, *Discourse production*, Edinburgh University Press, Edinburgh, 1978.

[Fillmore 1971]

C.J. Fillmore, *Some Problems for Case Grammar*, Geogetowm University Monograph Series of Languages and Linguistics 22. PP 35-36.

[Goldman 1975]

N.M. Goldman, *Sentence Paraphrasing from a Conceptual Base*, CACM 18, pp. 96-106.

[Halliday and Hansan 1976]

M.K.A. Halliday and R. Hansan, *Cohesion in English*, Longman, New York, 1976.

[Mann 1983]

W.C. Mann, *An Overview of the Nigel Text Generation Grammar*, Proceedings of the 21st Annual Meeting of the ACL, pp.79-84, 1983.

[Maudlin 1984]

M.L. Maudlin, *Semantic Rule Based Text Generation*, COLING 84, pp. 376-380, 1984.

[Mckeown 1979]

K.R. McKeown, *Paraphrasing Using Given and New Information in a Question-Answering System*, Prooceedings of the 17th Annual Meeting of the ACL, pp.67-72, 1979.

[Kuo 1989]

H.W. Kuo and J.S. Chang, *Systemic generation of Chinese Sentences*, ROCLING II, pp. 187-212, 1989.

[Li-Thomson 1982].

C.N.Li and S.A. Thompson, *Mandarin Chinese - A Functional Reference Grammar*, University of California Press, California, 1982.

[Patten 1985]

T. Patten, *A Problem Solving Approach to Generating Text from Systemic Grammars*, Proceedings of the 2nd Conference of the European Chapter of the ACL, pp. 251-257, 1985

[Tang 1975]

T.C.C. Tang, *A Case Grammar Classification of Chinese Verbs*, Hai-Guo Book Company, Taipei, Taiwan 1975.

[Woods 1970]

W.A. Woods, *Transition Network Grammars for Natural Langual Analysis*, Computational Linguistics 13, No.10, pp. 591-606, 1970.