Integration of Syntactic, Semantic and Contextual Information in Processing Grammatically Ill-Formed Inputs

Osamu Imaichi and Yuji Matsumo to
Graduate School of Information Science,
Nara Institute of Science and Technology
8916-5 Takayama, Ikoma, Nara 630-01, JAPAN
{osamu-im,matsu}@is aist-nara acjp

Abstract

This paper describes an integrated method for processing grammatically ill formed inputs We use partial parses of the input for recov ering from parsing failure In order to select partial parses appropriate for error recovery, cost and reward are assigned to them Cost and reward represent the badness and goodness of a partial parse, respectively The most appropriate partial parse is selected on the ba sis of cost and reward trade off The system contains three modules Module *A* handles local ill-formedness such as constraint violations Module *B* handles non-local ill formedness such as word order violations, and Module *C* handles non-local ill-formedness such as contextual ellipses These three modules work in a uniform framework based on the notions of cost and reward

1 Introduction

Interaction with computers using natural language has been a major goal of artificial intelligence Database sys tems and expert systems require flexible interfaces that allow users to communicate with the system in natu ral language Though many natural language processing systems have been developed, most of them assume that input sentences are grammatically correct How ever, when users communicate with the system, they often use grammatically ill-formed sentences, especially in spoken dialogues For example, the users omit some words, change the word order, or make some careless errors buch as agreement errors, misspellings or adding of extra words To use NLP systems in real applications, we need to construct an NLP system that can handle not only grammatically well-formed inputs but also grammatically *ill formed* inputs In other words, a robust NLP system should not just reject grammatically ill-formed inputs, but process them in any case

This paper describes a new method to deal with grammatically ill-formed inputs Our framework adopts a two-stage model consisting of ft *normal parsing process* and. a recovery proceed If the normal parsing process fails to find a complete parse for the input, the recov ery process is invoked Since we adopt *bottom up Chart parsing* [Kay, 1980] for the parsing engine, the recovery process can use partial parses for recovering from parsing failure The main mechanism is the selection of partial parses appropriate for error recovery In order to select the most appropriate one, *cost* and *reward* ' are assigned to partial parses Cost indicates the degree of inconsistency included in a partial parse, and it is calculated by *cost based unification* to be described below Reward indicates the degree of their contribution to error recovery This mechanism is used in Modules *A* and *B*

We also introduce the mechanism called *lexical ex pectation* Lexical expectation uses: the number and type information of the phrases with which the current phrase can combine in order to become complete Since we adopt a Japanese grammar based on HPSG (Head Driven Phrase Structure Grammar) [Pollard and Sag, 1994], this information is described in lexical entries This mechanism is used in Module *B*

111 formed inputs are handled by three modules de pending on the types of ill-formedness Module -4 han dies local ill formedness such as constraint violations Module *B* handles non-local ill-formedness such as word order violations, and Module *C* handles non local ill formedness such as incomplete sentential fragments and contextual ellipses Modules *A* and B try to find a phrase which would cover the whole input, whereas Module *C* tries to find an appropriate interpretation using contex tual information Each module alone is not sufficient for handling various types of ill-formedness The modules work on the basis of cost and reward trade off, and thus they are integrated into a uniform framework Conse quently, various typeB of ill-formedness can be handled in a flexible way

The paper is organized as follows Section 2 gives an overview of the related work and discusses the advan tages and disadvantages Section 3 reports the result of a corpus analysis of ill formed phenomena and classifies them Section *A* describes cost-based unification Section 5 gives the detailed description of Modules *A., B* and *C* Finally, section 6 summarizes the method and discusses the advantages and disadvantages

'The notions of cost and reward were inspired by Rele vance Theory [Sperberand Wiliion, 1986], though our notions are a bit different from those in their theory

2 Related works

In this section, we give an overview of the previous works and discuss their advantages and disadvantages

[Weischedel and Sondheimer, 1983] proposed the *relaxation method* based on ATN (Augmented Transition Network) [Woods, 1970] Their relaxation method uses *meta rules* A meta rule is a kind of production rule in the expert system and each of them describes a recovery method from a certain type of ill-formedness Meta rules are used when the normal parsing process has failed to find a complete parse The algorithm is as follows

> When the normal parsing process fails to find a complete parse for the input and is blocked, the recovery process applies meta rules in order to relax some constraints on the blocked parse and resume processing, and repeats the process until a complete parse is found

This method has the advantage of providing a uni form way to handle both syntactic and semantic ill formedness Moreover, this is an elegant method for extending the coverage of a grammar to include grammatically ill-formed inputs, while retaining a connection between acceptable expressions and unacceptable ones However, since they use an ATN parser, which works in a top-down and backtracking manner, the detection of errors is not always easy if the parsing failure does not occur at the erroneous position For the same reason, it is impossible to use the context to the right of the erroneous point in the input In addition, this method can handle only the types of ill formedness considered in advance If the system tries to handle various kinds of ill formedness using meta rules, the number of meta-rules becomes huge and the meta rules also conflict each other in the same way as production rules in expert systems

[Douglas and Dale, 1992J realized the relaxation method using the PATR-II grammar formalism [Shieber, 1986] They introduce *relaxation levels* and *relaxation packages* consisting of a set of constraints The relaxation packages are associated with each PATR-II gram mar rule and they represent sets of constraints which can be relaxed They are defined according to the relaxation levels If all relaxation packages at a relaxation level are satisfied, the grammar rule is used Otherwise, the violated relaxation packages are recorded, so that the constraints can be relaxed later

This approach describes constraints using *feature structure,* which is general linguistic data structure In this respect, their approach is superior to Weischedel and Sondheimer's work However, this method treats constraint violations only It cannot handle ill-formed phenomena that are not caused by constraint violations, such as word order violations To relax a constraint, the method simply removes the relaxation package, but the information indicating the constraint violation should be left for the further process Moreover, relaxation pack ages at each relaxation level must be described in ad vance The method does not consider the use of semantic and contextual information

A syntax-based approach is taken by [Mellish, 1989], who applied Chart parsing [Kay, 1980] to deal with gram matically ill formed inputs The advantage of using Chart parsing is that the information of partial parses can be used for processing grammatically ill formed in puts The aim of his work is to explore purely syntactic and grammar-independent techniques to enable a parser to recover from simple kinds of ill formedness such as unknown and misspelled words, omitted words and extra noise words The method is divided into two phases the bottom-up (BU) phase and the top-down (TD) phase The BU phase runs first to find a complete parse If the input is grammatically well formed, it successfully finishes at this phase Otherwise the parsing proceeds to the TD phase The TD phase is performed by hy pothesizing errors that may have caused a failure in the parsing In this system, edges in the Chart are assessed by a number of parameters to decide which edges may cause an error

The advantages of this method are that the normal parsing process is not affected by the extra mechanism for the recovery process and that the recovery process never repeats the same work done before by the bottom up parsing Moreover, owing to the property of a Chart parser, in contrast to an ATN parser, it is possible to use both left and right contexts in the input for the determi nation of the best parse This method efficiently parses grammatically ill-formed input which contains only one error However, if multiple errors occur in the input, the behavior gets dramatically worse because of the inefficiency of the top down phase This method uses only syntactic knowledge and can handle only simple types of errors It takes no account of the use of semantic and contextual information

The advantages and disadvantages of the previous works can be summarized as follows

The advantages

(1) to provide a uniform way to handle not only syntactic ill-formedness but also semantic ill formedness (Weischedel and Sondheimer's),

(2) to disallow the recovery process to affect the normal parsing process (Weischedel and Sondheimer's, Mellish's and Douglas and Dale's) and

(3) to be able to use the left and right contexts at erroneous points (Mellish's)

The disadvantages

(1) to work in a backtracking manner (Weischedel and Sondheimer's),

(2) to be unable to use the left and nght contexts at erroneous points (Weischedel and Sond heimer's),

(3) to consider the types of lll-formedness in advance (Weischedel and Sondheimer's and Dou glas and Dale's),

(4) to handle only simple syntactic errors (Mellish's) and

(5) to take no account of a recovery invoked by semantic and contextual information (Mellish's and Douglas and Dale's)

Our method for dealing with grammatically ill formed inputs overcomes the disadvantages while retaining the

Ill-formed phenomena	Telephone	Keyboard
Omission of words/phrases	619	514
Postpositional particles	113	26
Case elements	561	495
Speaker	445	328
Hearer	170	185
Topic	39	51
Extraneous words/phrases	644	4
Parentheses	10	1
Interjection	635	3
Self repairs	256	0
Wrong word order		
Inversion	5	0
Constraint violations	9	2
Ellipsis	21	69
At least one of the above	810	581
Total Number	1000	1000

Table 1 Our analysis of ill-formed sentences

advantages The use of a Chart parser allows us to obtain the advantage (3), and overcome the disadvantages (1) and (2) The two-stage model for processing grammatically ill formed inputs provides us with the advantage (2) Finally, the HPSG formalism and the integrated framework allows us to obtain the advantage (1) and overcome the disadvantages (3), (4) and (5)

3 Types of ill-formedness

We have analyzed types of ill formedness in 1000 Japanese sentences from both of keyboard and telephone dialogues in the ADD (ATR Dialogue Database) corpus [Ehara *et al*, 1990] The result of the analysis is presented in Table 1

On the basis of this analysis, we classify ill-formedness into the following types

Type 1 constraint violations

Type 2 structural violations

Type 3 incomplete sentential fragments

Type 1 includes omission of postpositional particles, and syntactic and semantic constraint violations Example 1 belongs to Type 1 The verb phrase "学校に行く" requires as its complement a noun phrase case marked by a postpositional particle, but the noun phrase "太郎" is not case marked In English, this type includes subject verb disagreement, case disagreement and so on

Example 1

"太郎　学校に　行く。
Taro　school TO-LOC　go
Taro goes to school

Type 2 includes word order violations such as inversion In example 2, the noun phrase "その本を" is inverted This type of ill-formedness has a syntactic structure that is not allowed by the grammar rules Since there is no grammar rule combining the verb phrase "私は読みました" with the noun phrase "その本を" in this order, it is impossible to handle the ill-formedness by relaxing constraints imposed on the grammar rules Instead, lexical information must be used

Example 2

"私は　読みました　その本を。
I-TOPIC/NOM　read PAST　the book ACC
I read the book

Type 3 includes ellipses and omission of case elements For example, an answer to a question is often a sentential fragment, usually a noun phrase, as in example 3 Such a sentential fragment is not understandable without contextual information For instance, without the context of the interrogative sentence it is impossible to understand that example 3 b means "太郎が来た (*Taro came*)"

Example 3

a　誰が　来たの？
who-NOM　came
Who came?

b　"太郎が。
Taro-NOM
Taro

In this paper, we are not concerned with extraneous words and phrases such as parentheses, interjection and self-repairs

Ill-formed inputs of Type 1 and Type 2 are processed without contextual information, whereas ill formed inputs of Type 3 must be processed using contextual information Type 1 is handled as a unification failure on grammar level but Type 2 and Type 3 require reference to lexical and contextual information In order to handle these types of ill formedness, we propose three modules Their details will be discussed in section 5

4 Cost-based unification

We extend unification operation to calculate the degree of inconsistency included in a partial phrase We call this extended unification *cost based unification*

Cost based unification has the following two properties

1 If the feature structures do not include inconsistent information, cost based unification behaves exactly the same as classical unification

2 If the feature structures include inconsistent information,
   (a) cost based unification *always* succeeds
   (b) the resultant feature structure receives a cost
   (c) the resultant feature structure maintains inconsistent information

A cost is assigned to the resultant feature structure according to the degree of inconsistency There are several possible ways to assign the cost The simplest way is to define the cost value as the number of inconsistent features We could also define a weight to each feature so as to estimate finer valuation of inconsistency For

the time being, we give a uniform weight to each feature The cost-based unification and the cost value are defined as follows

In classical unification, we cannot unify the following feature structures

$$\begin{array}{ll} \text{NUMBER} & \text{singular} \\ \text{PERSON} & \text{third} \end{array} \quad (i:$$

$$\begin{array}{ll} \text{NUMBER} & \text{plural} \\ \text{PERSON} & \text{third} \end{array} \quad (2;$$

However, in cost-based unification, we succeed in the unification, and the resultant feature structure is the following

$$\begin{array}{ll} \text{NUMBER} & \text{T\{singular, plural\}} \\ \text{PERSON} & \text{third} \end{array} \quad (3)$$

We call T{singular, plural} an *inconsistent set* The symbol T indicates an inconsistency An inconsistent set contains inconsistent values as its elements, such as plural and singular in the example above The cost value of the resultant feature structure is defined as the total number of excess elements in the inconsistency sets In this case, the value of the cost is 1

Next we consider how cost based unification works when an inconsistent feature structure and a consistent one are unified There are three types

The first type la the case where a new inconsistency is added by the unification In this case, the new inconsistent value is added to the inconsistent set For instance, given the feature structures (4) and (5) the resultant feature structure is as in (8) and its cost value is 2

$$\begin{array}{ll} \text{NUMBER} & \text{singular} \\ \text{PERSON} & \text{thud} \end{array} \quad (4)$$

$$\begin{array}{ll} \text{NUMBER} & \text{singular} \\ \text{PERSON} & \text{T\{first, second\}} \end{array} \quad <5)$$

$$\begin{array}{ll} \text{NUMBER} & \text{singular} \\ \text{PERSON} & \text{T\{first, second, third\}} \end{array}$$

The second type is the case where a new inconsistent value is already contained in the inconsistent set For instance, the cost-based unification of the feature structures (5) and (7) results in the feature structure (5) The cost value is 1

$$\begin{array}{ll} \text{NUMBER} & \text{singular} \\ \text{PERSON} & \text{second} \end{array} \quad (7)$$

The third type is the case where both feature structures contain inconsistency In this case, the result has the union of the inconsistent sets included in the feature structures For example, the resultant feature structure of cost-based unification of the feature structures (5) and (8) is equal to the feature structure (6) and its cost value is 2

$$\begin{array}{ll} \text{NUMBER} & \text{singular} \\ \text{PERSON} & \text{T\{second, third\}} \end{array} \quad (8)$$

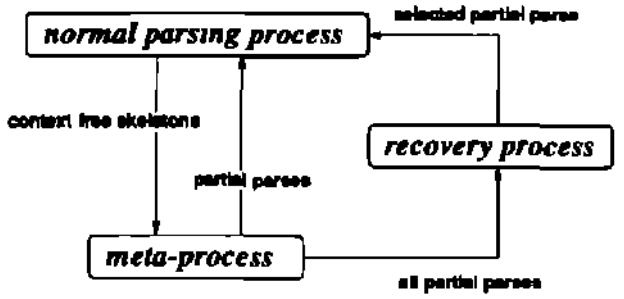

Figure 1 The flow of partial parses

5 Integrated method for dealing with grammatically ill-formed inputs

5 1 Outline

Figure 1 shows the architecture employed in our system and the flow of partial parses There are three components the *normal parsing -process,* the *meta process* and the *recovery process* The normal parsing process performs syntactic parsing only Chart parsing algorithm is used as the parsing engine This process only constructs context free skeletons and passes the intermediate results to the meta-process, which performs cost based unification The meta process returns them to the normal parsing process only when inconsistent information is not detected under cost based unification The meta process passes all the partial parses to the recovery process

If the normal parsing process succeeds in finding a complete parse, the recovery process is not invoked Otherwise, the recovery process is invoked by the meta-process The recovery process works according to Modules .4. and *B* to be discussed below It selects a partial parse appropriate for error recovery, and passes it to the normal parsing process Then the normal parsing process resumes to process This process finishes when the normal parsing process finds a phrase covering the whole input Then Module *C* is invoked for interpreting it within the context

5 2 Module *A*

If the normal parsing process fails to find a complete parse for the input, the recovery process is invoked and it selects a partial parse with a cost and passes it back to the normal parsing process, which then resumes its process The recovery process must therefore select a partial parse that gives most plausible explanation of the ill formedness We introduce the following criterion for the selection

- Selection criterion
  Select a partial parse with minimal cost and the maximal reward

The criteria of *reward* are defined as follows

- Reward criterion A
  1 Verb phrases
  2 Noun phrases
  3 Other phrases

- Reward criterion B

  1 Phrases which cover the widest range of the input string

  2 Rightmost phrases (in the case of Japanese [2])

The Reward criterion A indicates the ranking order of reward If the partial parse with the maximal reward is not decided by the Reward criterion A, the Reward criterion B is used For example, if two verb phrases exist, the partial parse with the maximal reward *is* selected using the Reward criterion B

Next we consider the trade-off between cost and reward If the recovery process selects a partial parse and some partial parses have the same cost, the phrases with the maximal reward are selected using the Selection criterion If some partial parses have the same reward the phrases with the minimal cost are selected The problem is which partial parse is selected, the partial parse with a smaller cost and a smaller reward or the partial parse with a larger cost and a larger reward In our current implementation, we prefer reward to cost Accordingly, we select the partial parse that has a larger reward

When Module $\mathcal{A}$ is invoked, it selects a partial parse which satisfies the Selection criterion and passes this to the normal parsing process Module $\mathcal{A}$ handles ill-formed inputs of Type 1

5 3 Module $\mathcal{B}$

If the selected partial parse has no cost but still does not cover the whole input sentence, Module $\mathcal{B}$ is invoked Module $\mathcal{B}$ uses the *lexical expectation* That is, it uses the *syntactic and semantic information* described in the feature structure of the phrases

In HPSG, a head has a SUBCAT list in head complement structures and an adjunct has a MOD value in adjunct-head structures The SUBCAT list specifies the phrases that the head requires as its complement The MOD value of an adjunct specifies the phrase which the adjunct requires as its head We use these SUBCAT information and MOD information for recovering from the ill formedness of Type 2

A phrase also has a CONTENT value For example, the CONTENT value of a verb shows the assignment of semantic roles, i e it encodes the verb's predicate-argument structure We could use the CONTENT value directly in order to recover from some types of ill-formedness However, the role assignment comes about by the structure sharing between a SUBCAT element's index and the value of some attribute (i e a semantic role) of the verb's CONTENT value Since the CONTENT value will be decided by the SUBCAT list or the MOD value we need not to use the CONTENT value directly, but can use it *indirectly* via the SUBCAT list or the MOD value

In the example 4, the verb phrase "私に貸して下さい" and the noun phrase "その本を" cannot be combined in this order by any grammar rule No partial parse, even with a cost, has been generated in the normal parsing process Accordingly, Module $\mathcal{A}$ is not effective This type of ill-formedness is solved by taking the lexical expectation into account The verb phrase "私に貸して下さい" has a SUBCAT list expecting subject and object noun phrases Module $\mathcal{B}$ tries to find the candidates for these phrases When a noun phrase is found, Module $\mathcal{B}$ checks the constraints that the verb imposes on the subject and object noun phrases, and it decides whether the found noun phrase is suitable for the subject or the object *In this example, since the noun phrase* "その本を" is suitable for the object of the verb, it fills in some semantic role of the verb and the verb phrase that covers the whole input is generated Then the generated partial phrase is passed to the normal parsing process

Example 4

"私に　貸して下さい、　その本を。
me DAT　please lend　the book ACC
Please lend me the book

The important points are the selection of the *seed* phrase and the search for phrases that the seed phrase requires The *seed* phrase triggers Module $\mathcal{B}$ and it plays a central role in the following process

1 Module $\mathcal{B}$ selects the seed phrase using the Selection criterion and the Reward criteria same as Module $\mathcal{A}$

2 When the seed phrase is selected, Module $\mathcal{B}$ looks at its SUBCAT list or its MOD value in order to search for the candidates which satisfy the requirement imposed by the seed phrase

3 Module $\mathcal{B}$ searches for the candidates according to the following Search criterion

- Search criterion
  Search for phrases that the seed phrase requires

If such a phrase is found, there are two cases depending on whether *the phrase is a complete phrase* [3] *or not*

If the found phrase is a complete phrase, Module $\mathcal{B}$ checks whether this phrase is syntactically and semantically suitable for the argument of the seed phrase During this process, if a partial parse with a cost comes out, Module $\mathcal{A}$ is invoked again

If the found phrase is incomplete, the phrase becomes the next seed phrase Module $\mathcal{B}$ is invoked recursively and it looks for phrases using the Search criterion

It is important to notice that the Selection criterion and the Reward criteria are used in both Modules $\mathcal{A}$ and $\mathcal{B}$ Modules $\mathcal{A}$ and $\mathcal{B}$ are defined separately, but they work in an interleaving manner If the selected partial parse has no cost but does not cover the whole input, Module $\mathcal{B}$ is invoked Otherwise, Module $\mathcal{A}$ is invoked Modules $\mathcal{A}$ and $\mathcal{B}$ thus work in a uniform framework based on the notion of cost and reward

5 4 Module $\mathcal{C}$

Modules $\mathcal{A}$ and $\mathcal{B}$ deal with the types of ill-formedness which are recoverable within the input and they try to find a phrase covering the whole input After Modules $\mathcal{A}$ *and* $\mathcal{B}$ *have processed the input, Module* $\mathcal{C}$ *receives* the result and finds an appropriate interpretation of the

[2] The background conception of this criterion *is* to select syntactic and semantic heads Since Japanese is a head final language, heads appear to the right of their complements in the most cases

[3] A phrase is complete if its SUBCAT list is empty

inputs using contextual information Module *C* handles ill-formed inputs of Type 3 and mainly does semantic processing

In example 3 b, the normal parsing process finds that "太郎が" is a complete noun phrase case marked by the postpositional subject case-marker "が" Modules *A* and *B* finish their processing because the phrase covers the whole input

We believe that such a fragment should not be re covered to form a complete sentence Rather, it should be understood within the context Therefore, Module *C* does not try to find the missing verb "来た" but checks whether the example 3 b IS understandable in the con text, that is, understandable as an answer to the question 3 a In the semantic representation of the question 3 a, the agent role of "来た" is not instantiated Since 郎が" is syntactically and semantically appropriate for the agent role of the verb, Module *C* regards the answer 3 b as well-formed in the context In this module, we will employ the context selection method proposed by Hirasawa [Hirasawa, 1995]

6 Conclusion

In this paper, we discussed a new method to deal with grammatically ill-formed inputs Our method overcomes the disadvantages of the previous works, whde still re taining their advantages It contains the following three modules

- Module *A* uses partial parses which have a cost,
- Module *B* uses *lexical expectation* mechanism, which uses syntactic and semantic information, and
- Module *C* uses contextual information for finding an appropriate interpretation of the input

Module *A* handles local ill formedness such as con straint violations Module *B* handles non-local ill formedness such as word order violations and Module *C* handles non-local ill-formedness such as incomplete sentential fragments and contextual ellipses Modules *A* and *B* try to find a phrase covering the whole input, whereas Module *C* receives the result of Modules *A* and *B* and finds an appropriate interpretation of the input using contextual information

We have integrated these modules into a uniform framework on the basis of the notion of cost and reward CoBt represents the degree of inconsistency included in a partial parse, and reward represents the goodness of partial parses Though each module alone is not enough to deal with various types of ill formedness, by integrat ing these modules, our framework can process them in a flexible way The role of each module is still explicitly separated although they are integrated into a uniform framework This makes it easy to extend the coverage of the framework

We did not discuss the types of ill-formedness such as self-repaired utterances, parenthetic phrases and repeti tions We need to deal with these types of ill-formedness especially when we handle spoken utterances Sagawa et al [Sagawa *et al*, 1994] propose a method for coping with such ill-formedness Their method transforms a self repaired utterance into well-formed one This method uses some linguistic clues, which include repetitions, un known words and isolated words Although we have not fully implemented, we will employ Sagawa's method as a pre processor for handling such ill-formedness

The objective of most of the works on robust pars ing is to construct a mechamsm for dealing with intra sentential Ul-formedness However, Japanese contains a lot of mter-sentential ill-formedness, such as contextual ellipsis, omission of case elements and zero pronoun, and to deal with these phenomena, we need to construct a module for handling intra-sentential ill-formedness In our framework, Module *C* plays this role At present, Module *C* is not implemented yet In future work, we intend to construct Module *C* on the basis of Relevance Theory [Sperber and Wilson, 1986] A part of the work is done by [Hirasawa, 1995]

A c k n o w l e d g e m e n t s

I am indebted to Dr Rnstiina Jokinen for helpful com ments on an earlier draft of this paper

R e f e r e n c e s

[Douglas and Dale, 1992] S Douglas and R Dale To wards Robust PATR In *COLING 92,* pages 468-174, 1992

[Ehara *et al*, 1990] T Ehara, N Inoue, H Kohyama, T Hasegawa, F Shohyama, and T Monmoto Con tents of the ATR Dialogue Database Technical Re port TR-I-0186, ATR, 1990

[Hirasawa, 1995] J Hirasawa. Contextual Interpretation of Utterances m HeWvance Theory Master's Thesis, NAIST IS MT351092 Nara Institute of Science and Technology, 1995

[Kay, 1980] M Kay Algorithm Schemata and Data Structure in Syntactic Processing Technical Report CSL-80 12, Xerox PARC, 1980

[Mellish, 1989] C S Mellish Some Chart based Tech niques for Parsing 111 Formed Input In *£7th Annual Meeting of ACL,* pages 102-109, 1989

[Pollard and Sag, 1994] C Pollard and I A Sag *Head Driven Phrase Structure Grammar* The University of Chicago Press, 1994

[Sagawa *et al*, 1994] Y Sagawa, N Onishi, and N Sugie A Parser Coping with Self Repaired Japanese Utterances and Large Corpus-based Evalu ation In *COLING 94,* pages 593-597, 1994

[Shieber, 1986] S M Shieber *An Introduction to Unification Based Approaches to Grammar* Number 4 in CSLI Lecture Notes CSLI, 1986

[Sperber and Wilson, 1986] D Sperber and D Wilson *Relevance* Blackwell, 1986

[Weischedel and Sondheimer, 1983] R M Weischedel and N K Sondheimer Meta-rules as a Basis for Pro cessing Ill-Formed Inputs *Computational Linguistics,* 9(3-4) 161-177, 1983

[Woods, 1970] W A Woods Transition Network Grammars for Natural Language Analysis *CACM,* 13(10) 591-606, 1970